



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY

A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

**TURBO KÓDY A JEJICH APLIKACE V PŘENOSOVÝCH
TECHNOLOGIÍCH**

TURBO CODES AND THEIR APPLICATIONS IN TRANSMISSION TECHNOLOGIES

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Michal Kuvík

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Pavel Šilhavý, Ph.D.

BRNO 2017



Bakalářská práce

bakalářský studijní obor **Teleinformatika**

Ústav telekomunikací

Student: Michal Kuvík

ID: 174340

Ročník: 3

Akademický rok: 2016/17

NÁZEV TÉMATU:

Turbo kódy a jejich aplikace v přenosových technologiích

POKYNY PRO VYPRACOVÁNÍ:

Nastudujte problematiku protichybového zabezpečení pomocí turbo kódů. Věnujte se problematice kódování a různým přístupům dekódování zpráv zabezpečených pomocí těchto kódů. Analyzujte uplatnění těchto kódů v perspektivních přenosových technologiích. U každé technologie analyzujte využitý kód, jeho typ, rozměry a metody dekódování. V programovém prostředí Matlab vytvořte nástroj umožňující vyhodnotit vlastnosti použitých kódů v jednotlivých technologiích v rozsahu dle dohody s vedoucím práce.

DOPORUČENÁ LITERATURA:

[1] Hanzo, L., Liew, T. H., Yeap, B. L., Turbo Coding, Turbo Equalisation and Space-Time Coding for Transmission over Fading Channels. JohnWiley, 2002, ISBN: 0470847263.

[2] Moon, T. K. Error Correction Coding: Mathematical Methods and Algorithms. Wiley-Interscience, 2005, ISBN-13: 978-0070010697.

[3] Lin, S., Costello, D. J.. Error Control Coding: Fundamentals and Applications, second edition, Prentice Hall: Englewood Cliffs, NJ, 2005 ISBN: 0-13-042672-5.

[4] Lin, S., Costello, D. J.. Channel Coding in Communication Networks: From Theory to Turbocodes, Wiley-ISTE, 2007 ISBN: 978-1-90520-924-8.

Termín zadání: 1.2.2017

Termín odevzdání: 8.6.2017

Vedoucí práce: Ing. Pavel Šilhavý, Ph.D.

Konzultant:

doc. Ing. Jiří Mišurec, CSc.
předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Cieľom tejto práce je objasniť problematiku protichybového zabezpečenia pomocou turbo kódov. Prácu by sme mohli rozdeliť na dve časti. V prvej časti sú teoreticky priblížené vlastnosti turbo kódov, ich kódovanie, rôzne prístupy dekódovania zabezpečených správ ako sú napríklad prístupy SOVA a MAP, taktiež analyzuje uplatnenie turbo kódov v perspektívnych prenosových technológiách. Druhá časť je zameraná na vytvorenie aplikácie pomocou programového prostredia Matlab a overenie vlastností vybraného prístupu v tomto programe.

KĽÚČOVÉ SLOVÁ

Turbo kódy, kóder, dekóder, protichybové zabezpečenie, SOVA, MAP, Matlab

ABSTRACT

Purpose of this work is to clarify problems of anti-error security with the help of turbo codes. This work can be divided into two parts. In the first part we look at the theoretical aspect of features of turbo codes, their coding and at various approaches of decoding secure messages for example SOVA and MAP, also at analyzing usage of turbo codes in promising transmission technologies. Second part is focused on creating an application with the help of software Matlab and on verifying attributes of this method in this software.

KEYWORDS

Turbo codes, encoder, decoder, forward error correction, SOVA, MAP, Matlab

KUVIK, Michal *Turbo kódy a jejich aplikace v prenosových technologiích*: bakalárska práca. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2016. 63 s. Vedúci práce bol Ing. Pavel Šilhavý, Ph.D

VYHLÁSENIE

Vyhlasujem, že som svoju bakalársku prácu na tému „Turbo kódy a jejich aplikace v prenosových technológiách“ vypracoval(a) samostatne pod vedením vedúceho bakalárskej práce, využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor(ka) uvedenej bakalárskej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto bakalárskej práce som neporušil(a) autorské práva tretích osôb, najmä som nezasiahol(-la) nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomý(-á) následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákoníka Českej republiky č. 40/2009 Sb.

Brno

.....

podpis autora(-ky)

POĎAKOVANIE

Rád by som poďakoval vedúcemu bakalárskej práce pánovi Ing. Pavlovi Šilhavému, Ph.D. za odborné vedenie, konzultácie, trpezlivosť a podnetné návrhy k práci.

Brno

.....

podpis autora(-ky)

OBSAH

Úvod	10
1 Kódovanie	11
1.1 Úvod	11
1.2 Konvolučné kódy	11
1.3 Blokové kódy	13
2 Turbo kódy	15
2.1 Úvod	15
2.2 Turbo kódovanie	16
2.3 Prekladanie	16
2.4 Zúženie	18
2.5 Turbo dekódovanie	19
2.5.1 Soft output viterbi algoritmus (SOVA)	20
2.5.2 Maximum a-posteriori algoritmus (MAP)	22
3 Využitie turbo kódov	24
3.1 Mobilné siete tretej generácie	24
3.2 CCSDS – Vesmírny komunikačný systém	26
3.3 Využitie turbo kódov v DVB	27
3.4 WiMAX	28
4 Implementácia v Matlabe	30
4.1 Hlavné menu	30
4.2 Systémové menu	31
4.3 Menu generovanie grafov	32
4.4 Blok kódovania	33
4.5 Blok dekódovania	36
5 Simulácie	38
5.1 Úvod	38
5.2 Vplyv veľkosti bloku	38
5.3 Vplyv zúženia	42
5.4 Vplyv dekódovacieho algoritmu	45
5.5 Vplyv počtu dekódovacích iterácií	50
5.6 Porovnanie výkonnosti systémov	53
6 Záver	56

Literatúra	57
Zoznam symbolov, veličín a skratiek	59
Zoznam príloh	62
A Obsah priloženého CD	63

ZOZNAM OBRÁZKOV

1.1	NRC kodér (2, 1, 3)	12
1.2	RSC kodér (2, 1, 3)	13
1.3	Kodér BCH (n, k, d_{\min}) s ($n - k$) posuvnými registrami.	14
2.1	Zapojenie turbokodéru.	16
2.2	Blokové prekladanie.	17
2.3	Náhodný prekladač.	18
2.4	Konvolučný prekladač.	18
2.5	Turbo konvolučný dekodér.	20
3.1	Turbo kodér používaný v CDMA.	25
3.2	Turbo kodér používaný v CCSDS.	27
3.3	Turbo kodér používaný v DVB.	28
3.4	Turbo kodér používaný vo WiMAXe.	29
4.1	Hlavné menu aplikácie.	30
4.2	Menu LTE systému.	32
4.3	Menu generovanie grafov.	33
4.4	Vývojový diagram kódovania.	35
4.5	Vývojový diagram dekódovania.	37
5.1	Výkonnosť LTE v závislosti na veľkosti bloku.	39
5.2	Výkonnosť CCSDS v závislosti na veľkosti bloku.	40
5.3	Výkonnosť WiMAX v závislosti na veľkosti bloku.	41
5.4	Výkonnosť LTE v závislosti na zúžení.	43
5.5	Výkonnosť CCSDS v závislosti na zúžení.	44
5.6	Výkonnosť WiMAX v závislosti na zúžení.	45
5.7	Výkonnosť LTE v závislosti na použitom dekódovacom algoritme.	47
5.8	Výkonnosť CCSDS v závislosti na použitom dekódovacom algoritme.	48
5.9	Výkonnosť WiMAX v závislosti na použitom dekódovacom algoritme.	49
5.10	Výkonnosť LTE v závislosti na počte dekódovacích iterácií.	51
5.11	Výkonnosť CCSDS v závislosti na počte dekódovacích iterácií.	52
5.12	Výkonnosť WiMAX v závislosti na počte dekódovacích iterácií.	53
5.13	Porovnanie výkonností testovaných systémov.	55

ZOZNAM TABULIEK

3.1	Aplikácia turbo kódov v komunikačných systémoch	24
5.1	Parametre simulácií - vplyvu veľkosti bloku dát	38
5.2	Parametre simulácií - vplyvu zúženia	42
5.3	Parametre simulácií - vplyvu dekódovacieho algoritmu	46
5.4	Parametre simulácií - vplyvu počtu dekódovacích iterácií	50
5.5	Parametre simulácií - vplyv voľby systému	54

ÚVOD

V roku 1993 Berrou, Glavix a Thitimajshima na konferencii Proceedings of IEEE International Communications zverejnili príspevok „*Near Shannon Limit Error-correcting Coding and Decoding: Turbo-codes*“ [1], tento príspevok prvýkrát predstavovala novú triedu konvolučných kódov, tzv. turbo kódy. Tieto kódy vykazujú veľký kódový zisk pri vysokých datových rýchlostiach a ich realizácia je relatívne jednoduchá. V článku bolo preukázané, že turbo kódy sú schopné zaistiť nízku bitovú chybovosť (BER) pri malých hodnotách odstupu signál/šum (SNR), ktoré sa blížia Shannonovmu limitu. Táto vzdialenosť od Shannonovho limitu môže byť iba 0,7 dB pre BER 10^{-5} . V nasledujúcich rokoch nastal výskum turbo kódov a došlo k jeho praktickému využívaniu.

Táto práca je zameraná na turbo kódy a ich aplikáciu v prenosových technológiách. Cieľom je naštudovať problematiku kódovania a dekódovania zabezpečených správ pomocou rôznych prístupov a vytvoriť aplikáciu v programovom prostredí Matlab, ktorá má slúžiť na overovanie vybraných prístupov. Turbo kódy by sme mohli rozdeliť do dvoch skupín, turbo blokové kódy a turbo konvolučné kódy. V prvej kapitole sú popísané konvolučné kódy a blokové kódy z ktorých turbo kódy vychádzajú. Druhá kapitola približuje turbo kódy, opisuje proces kódovania a dekódovania, popisuje kodér, dekodér a vysvetľuje princíp algoritmov SOVA a MAP, ktoré sú využívané pri dekódovaní. V tretej kapitole je popísané uplatnenie turbo kódov v jednotlivých prenosových technológiách s ich parametrami. Štvrtá kapitola sa venuje opisu programu vytvoreného v programovom prostredí Matlab, ďalej popisuje funkciu jednotlivých funkcií využitých pri simulovaní. Posledná kapitola komentuje výsledky vykonaných simulácií a popisuje vplyv jednotlivých parametrov na výkonnosť turbo kódov.

1 KÓDOVANIE

1.1 Úvod

Kódovanie je proces, ktorý sa používa vo všetkých komunikačných systémoch. Počas prenosu signálu informačným kanálom v reálnom živote dochádza z rôznych príčin k šumu. To znamená, že počas prenosu správy dôjde k chybe a prijatá správa je rozdielna od vyslanej. Preto vznikali systémy, ktoré zaistili detekciu vzniknutých chýb, či dokonca ich opravu. Tieto systémy by sme mohli rozdeliť do dvoch kategórií a to protichybové systémy ARQ (Automatic repeat request), ktoré dokážu chybu iba detekovať. Chyba je opravená opakovaním prenosu. Druhou kategóriou sú systémy FEC (Forward error correction), ktoré sú schopné chybu nielenže detekovať, ale následne aj opraviť bez opakovania prenosu. V tejto kapitole si priblížime blokové a konvolučné kódy. Tieto kódy sú využívané ako prostriedok realizácie v systémoch FEC [2].

1.2 Konvolučné kódy

Konvolučné kódy spracovávajú informačné bity sériovo, pričom vstupná dĺžka informačných blokov k je väčšinou malé číslo (1, 2, 3...), pričom výstupné n -bitové slová závisia nielen od aktuálnej k -tici informačných bitov, ale tiež aj na predchádzajúcich k -ticiach.

Konvolučné kódy by sme mohli rozdeliť do dvoch základných skupín a to nerekurzívne konvolučné kodéry (NRC) a rekurzívne konvolučné kodéry (RSC). Základným prvkom týchto kodérov je posuvný register, ktorý uchováva predchádzajúce hodnoty bitového toku a logické členy XOR. Tieto kodéry môžeme popísať tromi základnými parametrami (n, k, m) , kde n označuje počet výstupných bitov, k počet vstupov kodéra a m počet pamäťových buniek kodéru. Na obr. 1.1 vidíme zapojenie NRC kodéru s parametrami (2, 1, 2). Z týchto parametrov je možné zistiť informačnú rýchlosť (kódový pomer) R a dĺžku kódového ohraničenia K . Informačná rýchlosť udáva pomer počtu bitov vstupných a výstupných z kodéru a je definovaná ako [3]

$$R = \frac{k}{n}. \quad (1.1)$$

Dĺžka kódového ohraničenia udáva, ako dlho sa bit podieľa na procese kódovania [3]

$$K = m + 1. \quad (1.2)$$

Konvolučný kódér môžeme tiež popísať pomocou vytváracej matice, ktorá bude mať tvar

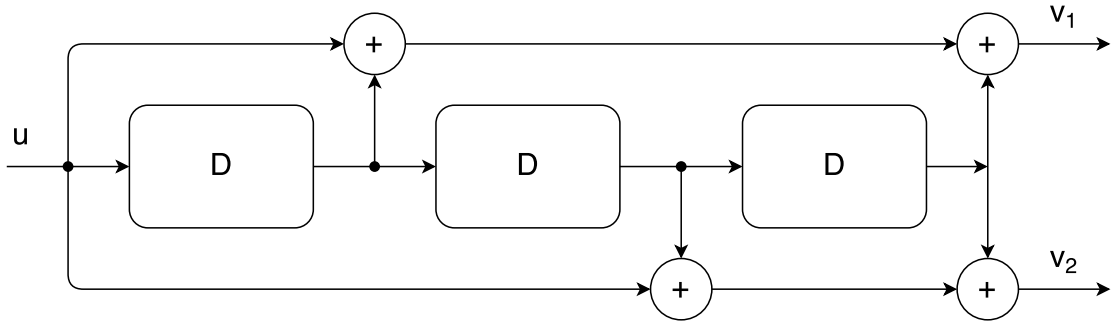
$$\mathbf{G}(D) = \begin{bmatrix} G_{(1)}^{(1)}(D) & G_{(2)}^{(1)}(D) \end{bmatrix}. \quad (1.3)$$

Vytváracie mnohočleny sa rovnajú nasledujúcim rovniciam, kde D nám označuje pamäťové bunky kódéra

$$G_{(1)}^{(1)}(D) = 1 + D + D^3, \quad (1.4)$$

$$G_{(2)}^{(1)}(D) = 1 + D^2 + D^3. \quad (1.5)$$

Z NRC kódéra môžeme odvodiť RSC kódér a to tak, že jeden zo zabezpečovacích výstupov privedieme na vstup, ako spätnú väzbu a druhý nastavíme ako systematický.

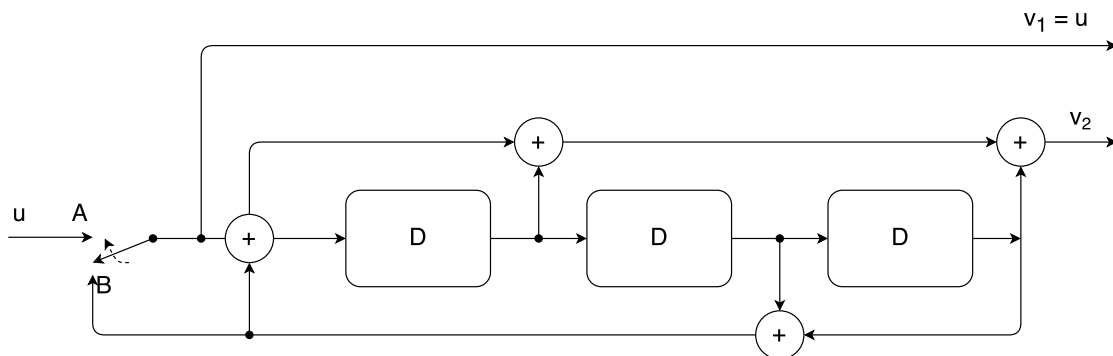


Obr. 1.1: NRC kódér (2, 1, 3)

Na obr. 1.2 môžeme vidieť RSC kódér odvodený z NRC kódéra na obr. 1.1. Kde u nám označuje vstup kódéra. Bity prichádzajúce na tento vstup bez akejkoľvek zmeny prechádzajú na výstup v_1 a sú označované ako systematické bity. Na výstupe v_2 sa nachádzajú bity paritné. D označuje pamäťové bunky a symbol \oplus XOR sčítanie modulo 2. Prepínač je v polohe A pri príchode vstupného kódového slova. Potom sa prepínač prepne do polohy B po dobu m krokov, kde m je počet pamäťových buniek kódéra. Týmto bude zaistené, že pamäťové bunky kódéra budú vynulované a ku vstupnej nezabezpečenej postupnosti je pridaných m zabezpečovacích bitov. Vytváracia matica pre tento kódér bude mať tvar

$$\mathbf{G}(D) = \begin{bmatrix} 1, \frac{G_{(1)}^{(1)}(D)}{G_{(2)}^{(1)}(D)} \end{bmatrix}, \quad (1.6)$$

kde $G_{(1)}^{(1)}(D)$ a $G_{(2)}^{(1)}(D)$ sú vytváracie mnohočleny NRC kódéra (rov. 1.4, 1.5). Konvolučné kódy môžu byť popísané aj grafickou formou a to stavovým, mriežkovým (trellis) alebo stromovým diagramom.



Obr. 1.2: RSC kodér (2, 1, 3)

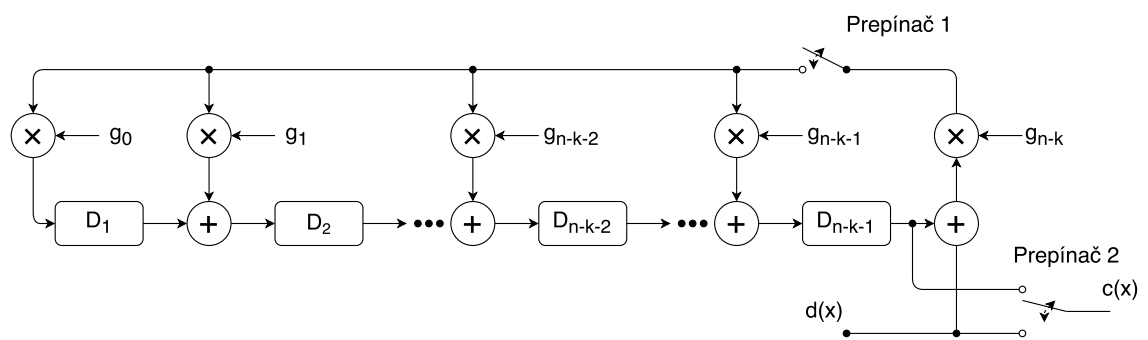
1.3 Blokové kódy

Hlavnou myšlienkou blokových kódov je vloženie takého množstva paritných bitov, aby bol dekodér schopný následne detekovať a opraviť chyby. Ich najdôležitejšia vlastnosť je fixná dĺžka kódového slova n . Môžeme ich rozdeliť na kódy systematické, to sú také, ktoré majú rozloženie informačných a zabezpečovacích miest v kódovom slove realizované podľa určitého systému a v každom slove rovnako. Druhou skupinou sú kódy nesystematické, konštrukcia týchto kódov nepredpokladá rozdelenie kódového slova na informačné a zabezpečovacie miesta.

Kódové slovo môžeme rozdeliť na k informačných bitov a r paritných (zabezpečovacích) bitov. Zabezpečovací proces je definovaný tzv. vytváracou (generujúcou) maticou alebo vytváracím polynómom. Príkladom blokových kódov definovaných vytváracím polynómom sú napr. BCH (Boseho-Chaudhuriho-Hocquenghemovy) kódy. BCH (n, k) , alebo tiež aj BCH (n, k, d_{\min}) , kde d_{\min} označuje minimálnu Hammingovu vzdialenosť. Hammingova vzdialenosť je definovaná, ako počet odlišných znakov v dvoch n znakových slovách [4]. BCH kódy sú definované vytváracím polynómom $g(x)$ (rov. 1.7). Tieto kódy môžu byť binárne alebo nebinárne, príkladom nebinárnych BCH kódov sú RS kódy.[5]

$$g(x) = g_0 + g_1x + g_2x^2 + \dots + g_{n-k-1}x^{n-k-1} + g_{n-k}x^{n-k}. \quad (1.7)$$

Polynómy sú zadávané v osmičkovej sústave $g(x)_{\text{octal}} = 23$, pretože ich dĺžka pri reálnom zápise je veľká. Následne ich je potrebné previesť do binárnej sústavy $g(x)_{\text{bin}} = 10011$ a potom na polynóm $g(x) = x^4 + x + 1$. Na obr. 1.3 [6] môžeme vidieť všeobecné zapojenie systematického kodéra pre BCH (n, k, d_{\min}) , kde \oplus je XOR, teda modulo 2 sčítanie a \otimes je násobenie.



Obr. 1.3: Kodér BCH (n, k, d_{\min}) s $(n - k)$ posuvnými registrami.

2 TURBO KÓDY

2.1 Úvod

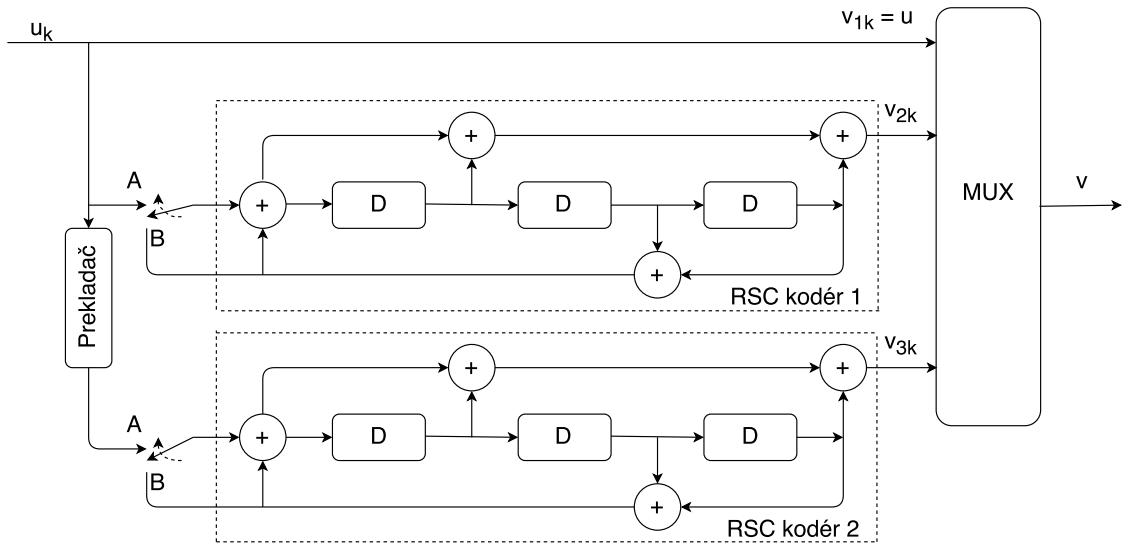
Claude Shannon vo svojom článku, ktorý bol zverejnený v roku 1949 prezentoval matematické základy pre komunikáciu pomocou kanálu za prítomnosti šumu a definoval *Shannon limit*. Tento limit nám udáva teoretickú maximálnu kapacitu pre daný informačný kanál a úroveň šumu. Pomocou konvolučných a blokových kódov pre detekciu chýb, ktoré boli stále prepracovanejšie sa darilo postupne približovať k tomuto teoretickému maximu. Bohužiaľ cenou za toto približovanie bola neustále narastajúca výpočtová náročnosť dekódovania a tieto kódy sa stávali nepraktickými. Ešte v roku 1990 bol rozdiel medzi teoretickou hranicou a praktickými implementáciami okolo 3 dB. Práve tento dôvod nútil hľadať nové kódy, ktoré mali schopnosť blížiiť sa Shannonovmu limitu. Riešením bola nová skupina kódov, ktoré boli pomenované Turbo kódy.

Claude Berrou, Alain Glavieux a Punay Thitimajshima, ktorí navrhli turbo kódy, v roku 1993 predstavili článok [1], v ktorom ukázali, že turbo kódy sa dokážu priblížiť ku Shannonovmu limitu iba na 0,7 dB pri bitovej chybovosti 10^{-5} . Bežne sa v turbo kódach používa kód konvolučný a to RSC. Ovšem používajú sa aj kódy blokove BCH, ktoré majú kódový pomer blížiaci sa jednej. Aj napriek tomu, že turbo kódy vychádzajú z konvolučných a blokových kódov dokážu čeliť vysokej výpočtovej zložitosti dekódovania a to tým, že na kódovanie používajú typicky dva alebo viac jednoduchších blokových či konvolučných kodérov a prekladač, ktorý je bližšie priblížený v kap. 2.3. Dekódovanie nastáva v iteráciách na základe pravdepodobnostného modelu a matematických princípov Bayesovskej štatistiky. Ovšem turbo kódy majú aj svoje nevýhody. Hlavnou nevýhodou je vysoká latencia, ktorú zapríčiňuje iteračný spôsob dekódovania a využívajú zložité dekódovacie algoritmy ako algoritmus SISO (Soft-Input Soft-Output) a MAP (Maximum A-Posteriori). Tiež nie sú presne dané optimálne parametre pre kodér a dekodér, čo sťažuje výber vhodných parametrov, ktorých je enormné množstvo možných kombinácií.

Od roku 1993 nastal intenzívny vývoj a výskum turbo kódov, dnes pod týmto názvom môžeme objaviť viac kódov a najpoužívanejším turbo kódom je kód, ktorý tvoria dva paralelne zretazené moduly. Vývoj týchto kódov pomohol k ich aplikácii do technológií ako napr. 3G, 4G mobilné siete, digitálna satelitná komunikácia alebo WiMAX [7].

2.2 Turbo kódovanie

Základná štruktúra turbo kodéru je znázornená na obr. 2.1. Vstupné bity sú označené ako u_k , bity na systematickom výstupe ako v_{1k} , bity na paritných výstupoch ako v_{2k} respektíve v_{3k} a výstupné multiplexované dáta ako v . Väčšinou tento kodér tvoria dva paralelne zapojené RSC kodéry, ktoré sú identické a medzi nimi je prekladač kap. 2.3. Výstupy kodérov môžu byť zúžené (puncturing) kap. 2.4 a následne sú multiplexované do výstupnej sekvencie $v = v_{10} + v_{20} + v_{30} + v_{11} + v_{21} + v_{31} \dots v_{1k} + v_{2k} + v_{3k}$.



Obr. 2.1: Zapojenie turbokodéru.

Tento turbo kodér má informačnú rýchlosť $1/3$ a vypočíta sa podľa vzťahu

$$\frac{1}{R_c} = \frac{1}{R_1} + \frac{1}{R_2} - 1. \quad (2.1)$$

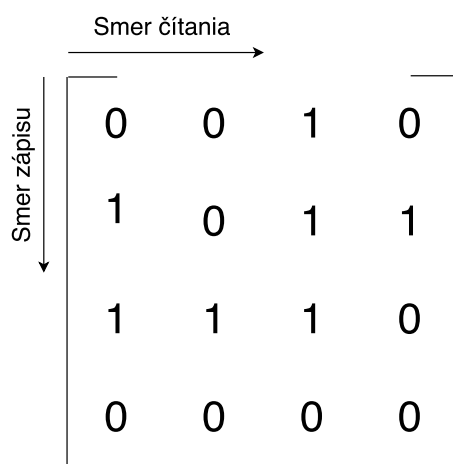
Informačná rýchlosť sa dá zmeniť pomocou zúženia a to tak, že systematické bity sa ponechajú a vynechané sú určité paritné bity. Ak by sme zúžili aj bity systematické znížili by sme výkonnosť kódu.

2.3 Prekladanie

Prekladanie (interleaving) je proces, pri ktorom dátovú sekvenciu preskupí prekladač (interleaver), čím zvýši Hammingovu váhu kódového slova. Hammingovu váhu je definovaná ako počet nenulových znakov v kódovom slove. Minimálna Hammingova váha je najmenšia hodnota Hammingovej váhy zo všetkých slov okrem slova s nulovými prvkami [2]. Tento proces napomáha pri korekcii skupinových chýb a zabezpečí, že jednotlivé kodéry neprodukujú rovnakú postupnosť. Chyby sú rozdelené

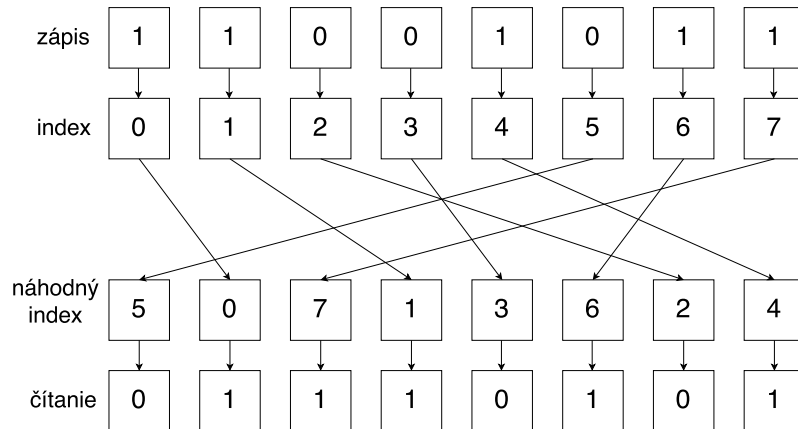
do viacerých blokov prenášanej postupnosti, čo zabraňuje prekročeniu zabezpečovacej schopnosti kódu. Kvalita prekladača súvisí s celkovou efektivitou turbo kódu a je jedným z hlavných faktorov, vďaka čomu sa približujeme Shannonovmu limitu. Prekladačov existuje veľké množstvo a každý má svoje špecifické vlastnosti, ktoré sú výhodné pre konkrétne komunikačné systémy. Najpoužívanějšími prekladačmi sú blokové, náhodné a tiež konvolučné prekladače.

Princíp blokového prekladača je, že vstupné bity sú zapisované do maticového tvaru po stĺpcoch a následne sú čítané po riadkoch, čo môžeme vidieť na obr. 2.2. Z obrázka vidieť, že zapísaná postupnosť bitov do matice bola $[0110 \dots 0100]$ a bity budú čítané v postupnosti $[0010 \dots 0000]$. Existuje viacero druhov blokových prekladačov a každý má trochu odlišný princíp, ktorý vyhovuje rozdielnym komunikačným systémom (obdĺžnikový blokový prekladač, diagonálny blokový prekladač...).



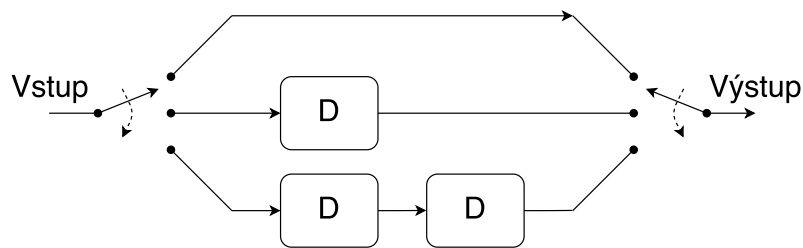
Obr. 2.2: Blokové prekladanie.

Druhým typom sú náhodné prekladače. Vstupné bity sú zapísané do prekladača a následne sú čítané na základe pseudonáhodnej sekvencie indexov. Princíp môžeme vidieť na obr. 2.3. Prekladač načítal postupnosť $[11001011]$ a na výstupe sme dostali náhodnú postupnosť $[01110101]$. Samozrejme aj v tejto skupine nájdeme viacero druhov prekladačov (pseudonáhodný prekladač, semi-random prekladač, uniformný náhodný prekladač...).



Obr. 2.3: Náhodný prekladač.

Konvolučný prekladač (obr. 2.4) sa skladá zo sady posuvných registrov, ktoré majú pevné oneskorenie. Do konvolučného prekladača vstupujú bity z ľavej strany a cyklicky sú pridelené do jednej z možných ciest. Vstupný a výstupný prepínač prekladača sú synchronizované, takže v prípade, že sú prepínače v prvej pozícii bit okamžite prechádza na výstup prekladača. V druhej pozícii je bit uložený do pamäti a prepustený na výstup až v ďalšom cykle, keď sa prepínače dostanú do rovnakej pozícii. Posledná pozícia znamená pre bit, že bude mať oneskorenie dva cykli, kým sa dostane na výstup. Spätný prekladač bude pracovať analogicky, avšak bit, ktorý nemal žiadne oneskorenie, bude mať oneskorenie dva cykli a naopak bit s oneskorením dva cykli nebude mať žiadne [8].



Obr. 2.4: Konvolučný prekladač.

2.4 Zúženie

Zúženie (puncturing) je proces, pri ktorom sa systematicky vymazávajú niektoré bity. Výhodou je zníženie šírky pásma, ktorá je potrebná na prenos informácie, avšak za cenu zníženia zabezpečovacej schopnosti kódu, čo je hlavnou nevýhodou [5]. Preto je potrebné urobiť kompromis medzi konečnou dĺžkou kódového slova

a zabezpečovacou schopnosťou kódu. Pri zúžení turbo kódov sa vypúšťajú len paritné bity, aby nedošlo k zníženiu výkonu kódu a to na základe zužovacej matice \mathbf{I} .

Ako príklad si ukážeme turbo kodér z obr. 2.1, jeho informačná rýchlosť je $R = \frac{1}{3}$, zúžením by sme chceli zvýšiť túto informačnú rýchlosť na $R = \frac{1}{2}$. Výstupné bitové postupnosti by boli $v_{1k} = [01100111]$, pre systematický výstup a pre paritné výstupy $v_{2k} = [00011101]$, $v_{3k} = [10001110]$. Táto matica by mala tvar

$$\mathbf{I} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

To znamená, že bity na miestach, kde sú v matici nuly by boli vypustené a bity na miestach jednotiek prepustené. Systematický výstup by bol bez zmeny a paritné výstupy by mali tvar $v_{2k} = [0010]$ a $v_{3k} = [0010]$. Pred dekódovaním je nutné na vymazané miesta doplniť logické nuly. Všeobecne môžeme výstup kodéra popísať ako $v = [v_{10}, v_{20}, v_{11}, v_{31}, v_{12}, v_{22}, \dots, v_{1k}, v_{3k}]$.

2.5 Turbo dekódovanie

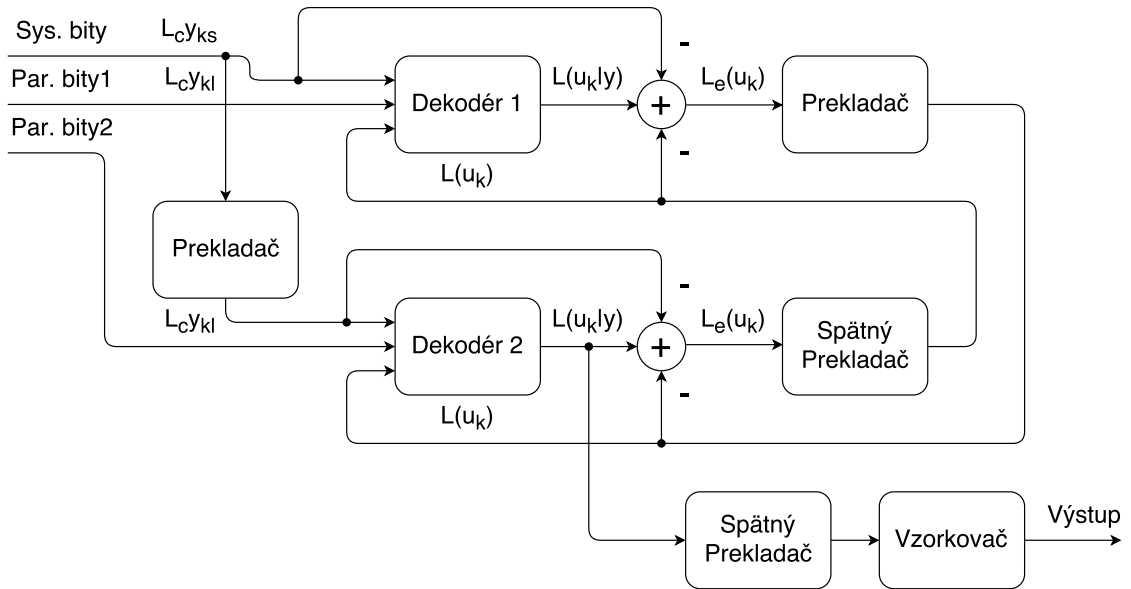
V prípade, že je použitý AWGN prenosový kanál je rozloženie porúch dané Gaussovou distribúciou to znamená, že prijatý signál nebude v tvare 0 a 1, ale bude pripomínať spojitý signál. Ak by sme tento signál porovnávali s určitou prahovou úrovňou, mohlo by dôjsť k nesprávnemu vyhodnoteniu logickej úrovne a tým by sa stalo dekódovanie neefektívne. Preto sa pri dekódovaní využíva tzv. mäkké výstupy LLR (*Log Likelihood Ratio*), hodnota a-posteriori pravdepodobností a znamienko pri tejto hodnote rozhoduje o dekódovanom bite. V praxi sa využívajú dva základné algoritmy, Viterbiho algoritmus SOVA (*Soft Output Viterbi Algorithm*) a druhý algoritmus MAP (*Maximum A-Posteriori*). Tieto algoritmy budú bližšie priblížené v nasledujúcich častiach tejto kapitoly.

Turbo dekodér obsahuje rovnaký počet konvolučných dekodérov, ako obsahuje turbo kodér konvolučných kodérov. Tiež je jeho súčasťou rovnaký interleaver ako bol použitý pri kódovaní a jemu príslušný deinterleaver. Na obr. 2.5, ktorý je prevzatý z lit. [3], môžeme vidieť zapojenie turbo dekodéra s popisom jeho vstupov a výstupov. Výstup z prenosového kanála je demultiplexovaný do postupnosti bitov, respektíve hodnôt, ktoré reprezentujú tieto bity, pretože boli ovplyvnené prenosovým kanálom. Systematické bity sa označujú ako $L_c y_{ks}$ a paritné bity obidvoch kodérov ako $L_c y_{kl}$, kde k označujú poradie bitu $k = 1, 2, 3 \dots N$. Do dekodéra tiež vstupuje vonkajšia informácia (a-priori) predchádzajúceho dekodéra $L(u_k)$, ktorá slúži k výpočtu a-posteriori informácie $L(u_k|y)$. V prípade, že výpočet vykonáva prvý dekodér, v prvej iterácii je hodnota a-priori informácie pre všetky bity nulová.

Aby a-posteriori informácia mohla slúžiť ako a-priori informácia pre nasledujúci dekodér je potrebné od nej odpočítať hodnotu systematického mäkkého vstupu $L_c(y_{ks})$ a a-priori informáciu z predchádzajúceho dekodéru $L(u_k)$ a to z dôvodu aby tieto hodnoty neovplyvňovali výpočet. Výpočet sa vykonáva podľa vzorca (2.2) [5].

$$L_e(u_k) = L(u_k|y) - L(u_k) - L_c y_{ks}. \quad (2.2)$$

V momente, keď si dekodéry predávajú a-priori informáciu, musí byť táto informácia pomocou interleaveru alebo deinterleaveru upravená tak, aby hodnoty pravdepodobností zodpovedali systematickému mäkkému vstupu vstupujúcemu do dekodéru. Tento postup výpočtu sa iteratívne opakuje, až kým nieje séria iterácií dokončená a následne sa určí tvrdý (hard) výstup z dekodéra. Znamienko a-posteriori informácie nám určuje tvrdý výstup + pre bit 1 a – pre bit 0.



Obr. 2.5: Turbo konvolučný dekodér.

2.5.1 Soft output viterbi algoritmus (SOVA)

V tejto kapitole si predstavíme SOVA algoritmus, ktorý ako už bolo spomenuté používa pri dekódovaní mäkký výstup v podobe a-posteriori pre každý dekódovaný bit. Nasledujúce body popisujú postup algoritmu a spôsob výpočtu jednotlivých krokov. Ešte pred samotným dekódovaním je potrebné dáta prenesené cez prenosový kanál demultiplexovať, aby zodpovedali postupnostiam bitov vysielaných z jednotlivých kódérov.

1. V prvok kroku je v každom úseku mriežkového grafu vypočítaná akumulovaná hodnota metriky danej cesty. Pre tento výpočet je v algoritme SOVA použitý

nasledujúci vzťah prebratý z lit.[9]

$$M_t^{(s)} = M_{t-1}^{(s)} + \frac{L_c}{2} u_t^{(s)} y_{t,1}^{(s)} + \frac{1}{2} \sum_{j=2}^N x_{t,j}^{(s)} y_{t,j}^{(s)} L_c + \frac{1}{2} u_t^{(s)} L(u_t), \quad (2.3)$$

kde

- $M_t^{(s)}$... je hodnota akumulovanej metriky v čase t a v stave s ,
- $u_t^{(s)}$... je vysielaný systematický bit,
- $x_{t,j}^{(s)}$... sú vysielané paritné bity,
- $y_{t,j}^{(s)}$... sú bity prijaté z prenosového kanála,
- $L(u_t)$... je a-priori hodnota pre bit u_t získaná z predchádzajúceho kódéra,
- L_c ... je spoľahlivosť prenosového kanála.

2. V prípade, že do jedného bodu (stavu) prichádzajú dve cesty sa pre výpočet akumulovanej metriky v čase $t+1$ využije vyššia hodnota. Zároveň sa vypočíta rozdiel týchto metrík Δ_t^s , ktorý nám určuje mieru spoľahlivosti, že sme urobili správne rozhodnutie. Môžeme uvažovať dve cesty \hat{c}_t^s a c_t^s smerujúce do stavu s v čase t , ktoré majú metriky $M(\hat{c}_t^s)$ a $M(c_t^s)$. Ak je cesta \hat{c}_t^s vybratá ako prežívajúca, pretože má vyššiu metriku, môžeme rozdiel týchto metrík definovať ako

$$\Delta_t^s = M(\hat{c}_t^s) - M(c_t^s). \quad (2.4)$$

3. Ďalší krok spočíva vo vyhľadani najpravdepodobnejšej cesty mriežkovým grafom a to tak, že v každom bode vyberieme najvyššiu hodnotu akumulovanej metriky a táto cesta sa berie ako najpravdepodobnejšia. Táto cesta nám reprezentuje dekódovanú bitovú postupnosť $\tilde{u} = (\tilde{u}_1, \tilde{u}_2 \dots \tilde{u}_k)$ a pre tieto bity sa bude počítat mäkký výstup dekodéru.
4. Pre ďalšie výpočty je potrebné zistiť tzv. update sekvencie pre každý úsek. Update sekvencie nám určuje, ktoré bity \tilde{u}_k by zmenili svoju hodnotu v prípade, že by bola ako prežívajúca cesta vybratá cesta konkurenčná. Hodnota 1 v update sekvencie určuje odlišnú hodnotu bitu, pričom najmenej významný bit LSB (Least Significant Bit) je na pravej strane. To znamená, ak by bol update sekvencie (11001), bity \tilde{u}_1 , \tilde{u}_2 a \tilde{u}_5 by zmenili svoju hodnotu. V prípade, že prechádzame mriežkový graf z ľavej strany narazíme na pozíciu k , kde sa zlučuje najpravdepodobnejšia cesta s cestou konkurenčnou. Hodnota tohoto bitu v update sekvencie je stále 1 a označujeme ho, ako najviac významný bit MSB (Most Significant Bit). Druhý najvýznamnejší je bit u_{k-1} , až kým sa nedostaneme k bitu LSB.

5. Pomocou hodnôt updatu sekvencie vypočítame a-posteriori $L(u_k|y)$ pre jednotlivé bity a to pomocou nasledujúceho vzorca prebratého z lit.[5]

$$L(u_k|y) = \tilde{u}_k \cdot \min \Delta_t^s. \quad (2.5)$$

Ako prvé je potrebné z updatu sekvencie určiť hodnotu minimálneho rozdielu metrík $\min \Delta_k^s$, ktorá je daná ako minimálna hodnota rozdielu metrík zlučujúcich ciest pozdĺž najpravdepodobnejšej cesty. Ako príklad si ukážeme, keď budeme určovať minimálny rozdiel metrík pre bit \tilde{u}_5 . V update sekvencie pre tento bit si zoberieme MSB, ktorý je vždy 1, následne zistíme druhý MSB bit z updatu sekvencie pre bit \tilde{u}_{5+1} a tretí MSB pre bit \tilde{u}_{5+2} . Takto postupujeme až kým sa nedostaneme ku bitu \tilde{u}_k . Hodnota bitu 1 v sekvencii týchto bitov nám určuje, ktoré hodnoty rozdielu metrík pripadajú v úvahu a následne je z týchto hodnôt vybratá minimálna hodnota. Táto hodnota nám určuje minimálnu hodnotu rozdielu metrík pre bit \tilde{u}_5 . Týmto postupom sme schopný zistiť a-posteriori pravdepodobnosť pre dekódovanú postupnosť a v nasledujúcom kroku vypočítať a-priori pravdepodobnosť pre ďalší dekodér.

6. A-posteriori pravdepodobnosť je možné rozložiť do troch častí pričom platí vzorec (2.6) [5]

$$L(u_k|y) = L(u_k) + L_c y_{t,1} + L_e(u_k), \quad (2.6)$$

kde

- $L(u_k)$... je a-priori hodnota získaná z predchádzajúceho kodéra,
- $L_c y_{t,1}$... je systematická hodnota prijatá z prenosového kanála, ktorá je vynásobená parametrom L_c ,
- $L_e(u_k)$... je konečná hodnota produkovaná dekodérom.

Po úprave tohoto vzorca dostávame vzťah pre výpočet a-priori pre nasledujúci dekodér

$$L_e(u_k) = L(u_k|y) - L(u_k) - L_c y_{t,1}. \quad (2.7)$$

Výstupné hodnoty sú následne interleavnuté a prichádzajú na vstup druhého dekodéra, kde sa postup opakuje, až kým sa nedosiahne finálny počet iterácií.

2.5.2 Maximum a-posteriori algoritmus (MAP)

Druhým algoritmom, ktorý sa využíva pri dekódovaní turbo kódov je tzv. MAP algoritmus. Rovnako ako algoritmus SOVA pri dekódovaní dátovej postupnosti využíva mäkký výstup tzv. a-posteriori pravdepodobnosť. Dekódovanie spočíva v dvojitom prechode trellis diagramom a to v doprednom smere a následne v spätnom smere. Pri

týchto prechodoch diagramom sa v jednotlivých uzloch vypočítajú stavové pravdepodobnosti $\alpha_t^{(s)}$ a $\beta_t^{(s)}$. V nasledujúcich bodoch si popíšeme kroky algoritmu a výpočty, ktoré sú v týchto krokoch vykonávané.

1. V prvom kroku je vypočítaná stavová pravdepodobnosť $\alpha_t^{(s)}$ v doprednom smere. Táto pravdepodobnosť sa vypočíta pomocou nasledujúceho vzorca, ktorého odvodenie je možné vyhľadať v lit.[10]

$$\alpha_t^{(s)} = \sum_{j=0}^1 \alpha_{t-1}^{b(j,s)} \gamma_{t-1}^{b(j,s)}, \quad (2.8)$$

kde

$\alpha_t^{(s)}$... je stavová pravdepodobnosť v čase t pre stav s ,
 $\alpha_{t-1}^{(s)}$... je stavová pravdepodobnosť v čase $t - 1$ pre stav s ,
 $\gamma_{t-1}^{(s)}$... je pravdepodobnosť prechodu z času $t - 1$ do času t .

Pre výpočet pravdepodobnosti prechodu je použitý vzorec 2.9, ktorého odvodenie je v lit.[10]

$$\gamma_t^{b(j,s)} = L(u_t) \exp(u_t y_{t,1} + x_t y_{t,2}), \quad (2.9)$$

kde

$L(u_t)$... je a-priori pravdepodobnosť z predchádzajúceho dekodéru,
 u_t, x_t ... je vysielaná dvojica bitov (systematický a paritný),
 $y_{t,1}, y_{t,2}$... je priatá dvojica bitov (systematický a paritný).

2. Tento krok spočíva vo vypočítaní stavovej pravdepodobnosti $\beta_t^{(s)}$ v spätnom smere, pre ktorý platí nasledujúci vzorec [10]

$$\beta_t^{(s)} = \sum_{j=0}^1 \beta_{t+1}^{f(j,s)} \gamma_t^{f(j,s)}, \quad (2.10)$$

kde jednotlivé symboly majú rovnaký význam ako vo vzorci 2.8, len β nám označuje spätnú stavovú pravdepodobnosť.

3. Po vypočítaní týchto hodnôt sme schopný určiť hodnotu mäkkého výstupu pomocou vzorca 2.11, prebratého z lit.[10]

$$L(u_k|y) = \log \left[\frac{\sum_s \alpha_t^{(s)} \gamma_t^{(1,s)} \beta_{t+1}^{f(1,s)}}{\sum_s \alpha_t^{(s)} \gamma_t^{(0,s)} \beta_{t+1}^{f(0,s)}} \right] \quad (2.11)$$

Tento mäkký výstup je možné rozložiť rovnako ako pri SOVA algoritme a to podľa vzorca 2.6 a následne podľa vzorca 2.7 dopočítať a-priori pravdepodobnosť pre nasledujúci dekodér.

Keďže je MAP algoritmus veľmi náročný na výpočty, nezvykne sa využívať v iteračných dekodéroch turbo kódov. Využívajú sa jeho modifikácie ako Log-MAP algoritmus, ktorý pracuje v logaritmicknej oblasti, kde je násobenie prevedené na sčítanie a tým sa výrazne zredukuje jeho výpočtová náročnosť.

3 VYUŽITIE TURBO KÓDOV

V súčasnosti majú turbo kódy veľké množstvo aplikácií v komunikačných systémoch a to vďaka svojej výkonnosti. Ako bezpečnostné kódovanie sa využívajú pri drôtovej alebo bezdrôtovej komunikácii, pri ukladaní dát, pri komunikácii s objektami vo vesmíre a v ďalších aplikáciách, kde je potrebné dosiahnuť spoľahlivého prenosu. V tabuľke 3.1 môžeme vidieť najznámejšie aplikácie turbo kódov z ich parametrami, tabuľka je prevzatá z lit.[11, 12, 13, 14, 15, 16, 17, 18].

Tab. 3.1: Aplikácia turbo kódov v komunikačných systémoch

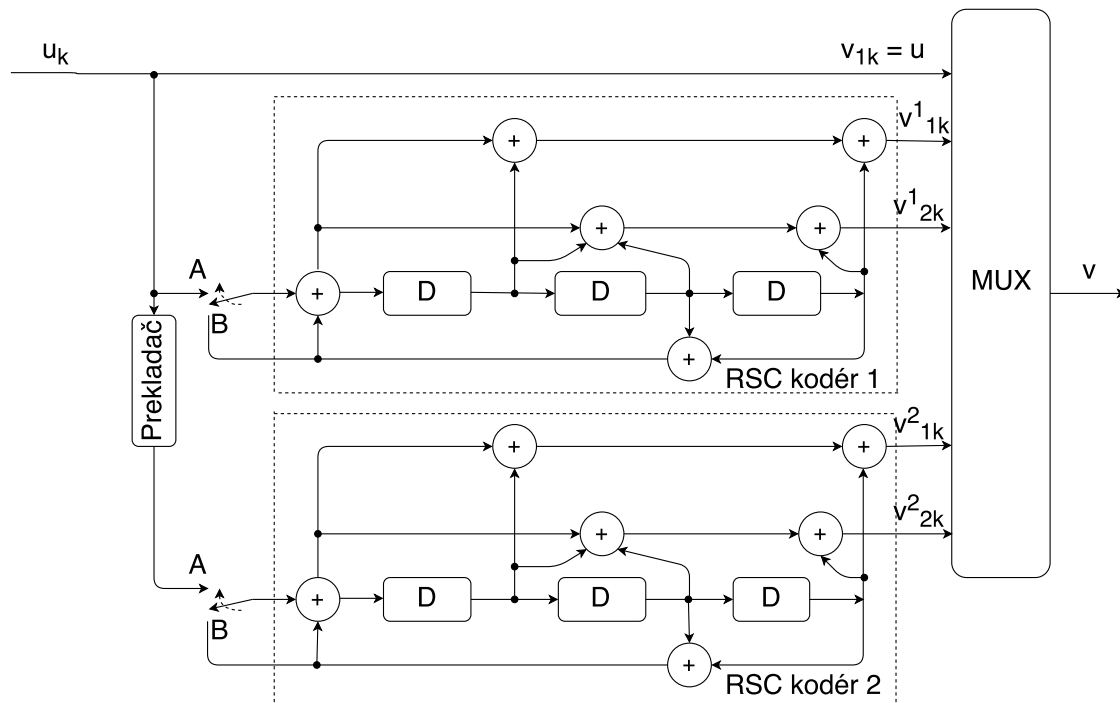
Aplikácia	Turbo kód	Polynómy	Kódový pomer
UMTS (3GPP)	binárny, 8-stavový	13, 15	1/3
CDMA2000 (3GPP2)	binárny, 8-stavový	13, 15, 17	1/5, 1/4, 1/3, 1/2
LTE (3G PP LTE)	binárny, 8-stavový	13, 15	1/3, 1/2
CCSDS (deep space)	binárny, 16-stavový	23, 33, 25, 37	1/6, 1/4, 1/3, 1/2
DVB-RCS (return channel)	duo-binárny, 8-stavový	13, 15	1/3 až do 6/7
DVB-RCT (return channel)	duo-binárny, 8-stavový	13, 15	1/2, 3/4
IEEE 802.16 (WiMAX)	binárny, 8-stavový	15, 13, 11	1/5, 1/3, 1/2
IEEE 802.16e (Mobile WiMAX)	duo-binárny, 8-stavový	15, 13, 11	
Eutelsat (Skypeplex)	duo-binárny, 8-stavový	15, 13	4/5, 6/7

3.1 Mobilné siete tretej generácie

V tejto časti kapitoly si priblížime mobilné siete tretej generácie, ktoré pre zabezpečenie prenášaných dát používajú turbo kódy. Medzi tieto systémy patrí UMTS (Universal Mobile Telecommunication System) a LTE (Long Term Evolution), ktoré sú štandardizované skupinou telekomunikačných asociácií 3GPP (Third Generation Partnership Project) a systém CDMA (Code Division Multiple Access), ktorý štandardizovala skupina 3GPP2 (Third Generation Partnership Project 2).

Medzi dva najrozšírenejšie 3G mobilné systémy patrí systém UMTS, ktorý je ďalším stupňom vývoja GSM sietí. Pri UMTS sa pre zabezpečenie prenášaných dát používajú konvolučné kódy a tiež turbo kódy. Nasledujúce parametre využívané pri kódovaní sú prebraté z lit.[12] a platia pre turbo kódy. UMTS ako kodér používa 8-stavový RSC kodér, ktorého vytváracie polynómy v osmičkovej sústave sú 13 a 15, kde 13 je spätná väzba kodéru. Kódový pomer turbo kodéru je $1/3$ s kódovým obmedzením $K = 4$. Veľkosť blokov dát sa volí v rozmedzí 40 až 5114 bitov. UMTS využíva interný interleaver. Skladá sa z bloku dát, ktorý je zapísaný do obdĺžnikovej matrici a následne sú bity preskupené podľa predpísaného algoritmu. Tento algoritmus je presne popísaný v lit.[12].

Druhým 3G systémom je CDMA, taktiež používa 8-stavový RSC kodér (obr.3.1), ktorého vytváracie polynómy v osmičkovej sústave sú 13, 15, 17, kde 13 je polynóm spätnej väzby. Hlavný rozdiel medzi UMTS a CDMA je v prekladaacom algoritme, povolenom rozsahom veľkosti blokov a kódovým pomerom. Povolené veľkosti blokov dát sú od 186 po 20730 bitov, presné hodnoty týchto blokov je možné vyhľadať v lit.[13]. Kódový pomer využívaný pri turbo kódach je $1/5$, $1/4$, $1/3$ a $1/2$. Interleaver pracuje tak, že blok dát je sekvenčne zapisovaný do poľa v sekvencii adries a následne je celá sekvencia čítaná podľa určenej postupnosti adries. Túto určenú postupnosť adries je možné vyhľadať v lit.[13].



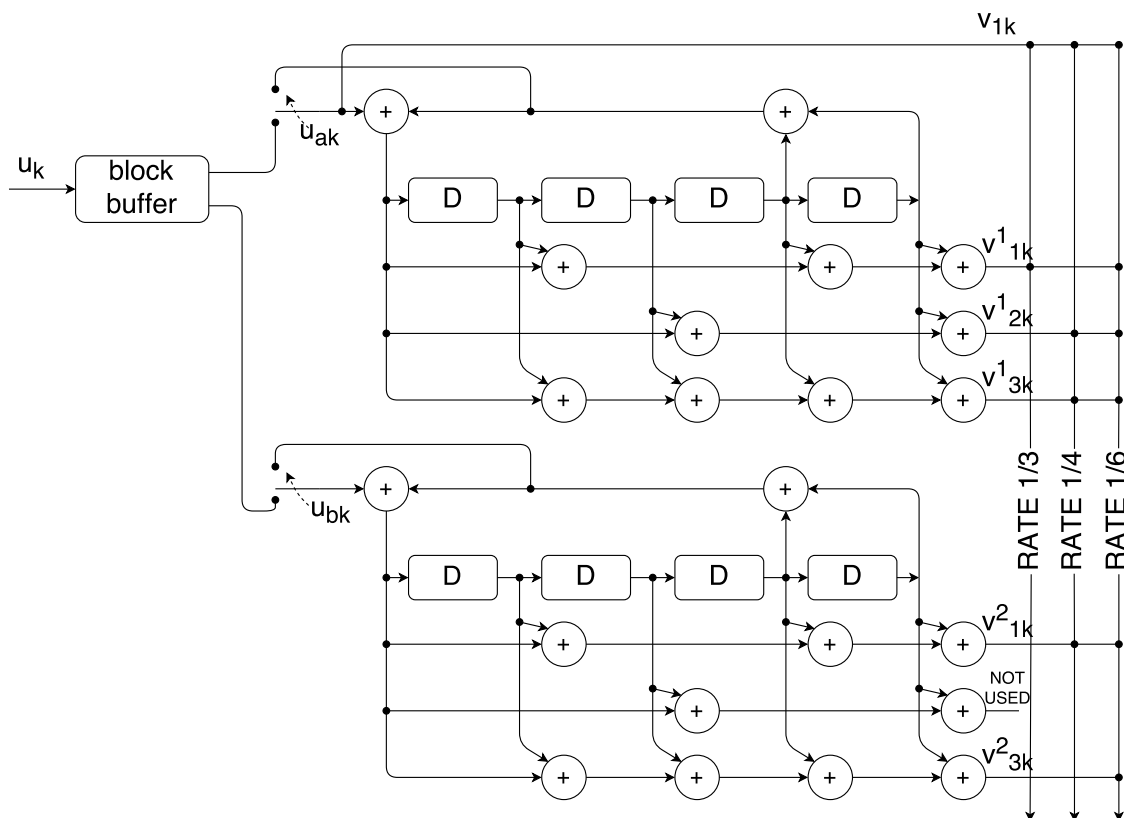
Obr. 3.1: Turbo kodér používaný v CDMA.

Ďalšou technológiou, ktorá spadá do štandardu 3G je vysokorýchlostný internet v mobilných sieťach GSM označovaný ako LTE. LTE je niekedy nesprávne označovaný ako 4G LTE, pričom nespĺňa podmienky ITU pre siete štvrtej generácie. Štvrtú generáciu dátových sietí predstavuje až štandard LTE Advanced. LTE využíva pre zabezpečenie dát turbo kódy a tiež konvolučné kódy. Ako pri predchádzajúcich systémoch používaný je 8-stavový RSC kodér, ktorého vytváracie polynómy v osmičkovej sústave sú 13 a 15 s kódovým pomerom $1/3$ a podporované veľkosti blokov dát od 40 do 6144 bitov. Interlaver má pre každý bit z bloku dát určené špecifické parametre, podľa, ktorých vypočíta výstupný index. Informácie o LTE boli prebraté z lit.[14], kde je možné vyhľadať aj tabuľku pre interleaver a výpočet výstupného indexu bitu.

3.2 CCSDS – Vesmírny komunikačný systém

Konzultačný výbor pre systémy spracovávajúce vesmírne dáta (CCSDS), bol založený v roku 1982 a jeho hlavnou úlohou bolo stanovenie medzinárodných štandardov pre vesmírnu komunikáciu so vzdialenými kosmickými sondami vo vesmíre. Tento systém používajú agentúry z celého sveta (NASA, RSA, JAXA, ESA...) a v roku 1999 boli pre zabezpečenie prenosu dát zvolené turbo kódy, ako nová možnosť.

Systém využíva 16-stavový binárny RSC kodér (obr.3.2), ktorého vytváracie polynómy sú 23, 33, 25 a 37, kde 23 tvorí spätnú väzbu kodérov. Využívané kódové pomery sú $1/6$, $1/4$, $1/3$, $1/2$ a veľkosť blokov dát je volená z hodnôt 1784, 3568, 7136, 8920 alebo 16382 bitov. Interleaver v CCSDS pracuje pomocou permutácii bitu za bit pre celý blok dát a výstupný index sa vypočíta algoritmom, ktorého parametre sú špecifické pre rôzne dĺžky blokov dát. Obrázok kodéra, parametre ako aj podrobný popis algoritmu pre interleaver s jeho parametrami, sú prebraté z lit.[15]

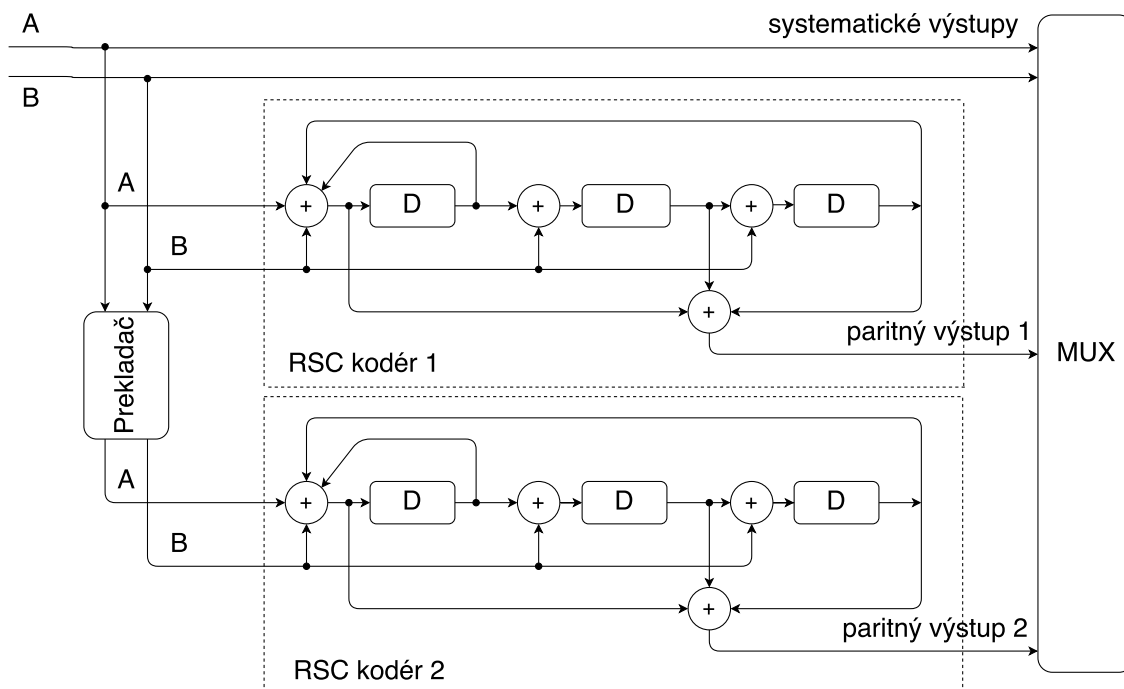


Obr. 3.2: Turbo kodér používaný v CCSDS.

3.3 Využitie turbo kódov v DVB

DVB (Digital Video Broadcasting) je skupina štandardov pre vysielanie digitálnej televízie spravovaný konzorciom DVB Project. Existuje veľa druhov DVB, ktoré sa odlišujú spôsobom šírenia signálu. Bližšie si priblížime DVB-RCS, ktorý sa zaoberá spätným kanálom komunikácie cez satelit a DVB-RCT, ktorý špecifikuje spätný kanál pre pozemnú digitálnu televíziu. Vzhľadom na to, že tieto systémy majú malý priestor na uplink je potrebné spoľahlivé protichybové zabezpečenie.

Obidva štandardy využívajú rovnaký turbo kodér (obr. 3.3) a to duo-binárny, 8-stavový, ktorého vytváracie polynómy sú 13 a 15, kde 15 je polynóm spätnej väzby. Rozdiel je v podporovaných dĺžkach dátových blokov a kódovom pomere, kým DVB-RCS podporuje bloky od 96 až po 1728 bitov a kódový pomer od 1/3 až do 6/7, pri DVB-RCT sú použité dátové bloky od 144 do 648 bitov a kódový pomer 1/2, 3/4. Informácie o DVB-RCS a DVB-RCT ako aj obr. 3.3 sú prebraté z lit.[16, 17].

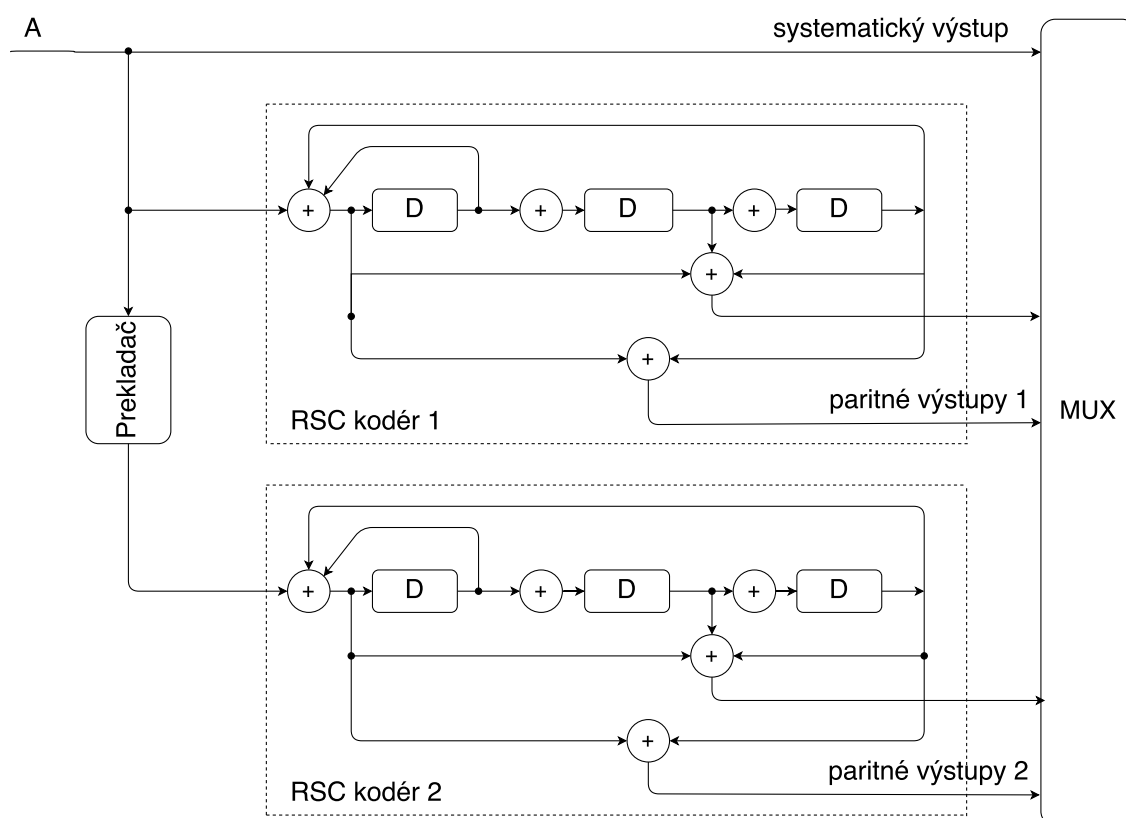


Obr. 3.3: Turbo kódér používaný v DVB.

3.4 WiMAX

WiMAX (Worldwide Interoperability for Microwave Access) je bezdrôtová technológia definovaná v norme IEEE 802.16. Rozšírením tejto základnej normy na normu IEEE 802.16e prinieslo podporu mobility, označuje sa ako Mobile WiMAX.

WiMAX používa 8-stavový binárny turbo kódér (obr.3.4), ktorého vytváracie polynómy v osmičkovej sústave sú 11, 13, 15, kde 15 je polynóm spätnej väzby. Kódový pomer sa určuje na základe použitej modulácie a je v rozmedzí od $1/5$ po $1/2$. Bloky dát sú volené od 12 po 1152 bitov. Interleaver určuje adresu bitu na základe veľkosti bloku a vypočíta ju podľa špeciálneho algoritmu. Presný postup algoritmu je možné vyhľadať v lit.[18], odkiaľ sú prebraté informácie o WiMAXe.



Obr. 3.4: Turbo kodér používaný vo WiMAXe.

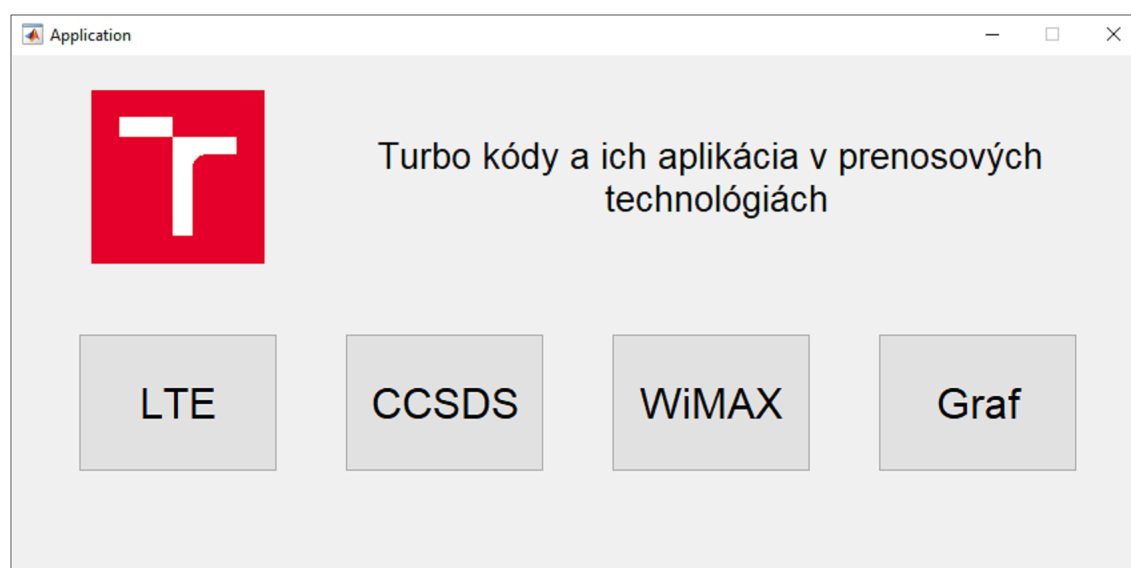
4 IMPLEMENTÁCIA V MATLABE

Prvou časťou bakalárskej práce bolo teoreticky popísať princípi a vlastnosti turbo kódov a tiež systémy, ktoré tieto kódy využívajú. Úlohou druhej praktickej časti je vytvorenie programu a následná simulácia vytvorených systémov za účelom overenia ich vlastností a výkonnosti. Pre tieto potreby bol vytvorený program v programovom prostredí Matlab verzie 2016a.

Program je možné rozdeliť do troch hlavných GUI (Graphical User Interface). Prvú časť tvorí hlavné menu aplikácie, funkcia `Application.m`. Druhou je menu pre jednotlivé systémy, funkcie `LTE_App.m`, `WiMAX_App.m` a `CCSDS_App.m`. Poslednú časť tvorí menu pre generovanie grafov z uložených súborov, funkcia `Graf_App.m`. V nasledujúcom texte bude taktiež popísaný proces kódovania a dekódovania, ktoré sú využívané v týchto funkciách. Ďalej budú popísané všetky funkcie, ktoré sa na týchto procesoch podieľajú, ich funkcia, vstupné a výstupné premenné. Keďže všetky tri systémy využívajú funkcie, ktorých význam je rovnaký, avšak sú upravené pre daný systém, budú tieto funkcie popísané len pre systém LTE.

4.1 Hlavné menu

Po spustení aplikácie sa zobrazí hlavné menu, ktoré je zobrazené na obr. 4.1. V menu je možné stlačením tlačítka zvoliť systém, ktorý chceme simulovať, alebo vybrať štvrtú možnosť a to generovanie grafov z uložených súborov. Po vybratí jednej z možností sa hlavné menu automaticky zatvorí a otvorí sa zvolené menu.



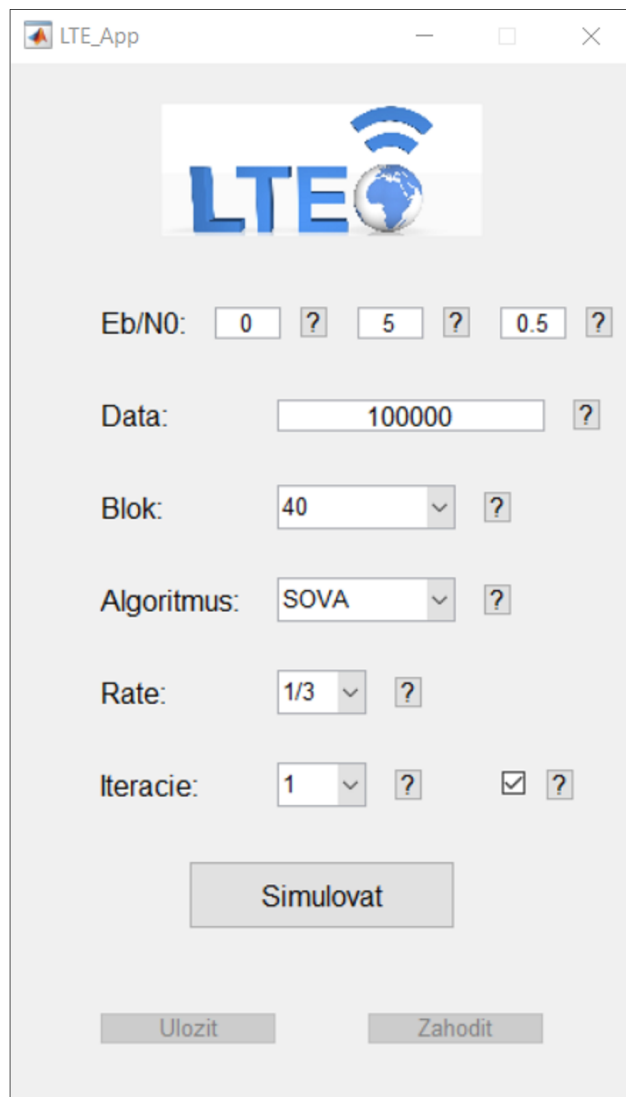
Obr. 4.1: Hlavné menu aplikácie.

4.2 Systémové menu

Na obr. 4.2 je zobrazené menu LTE systému, ktoré sa zobrazí po zvolení LTE systému v hlavnom menu. Ostatné systémy majú rovnaké polia pre zadávanie parametrov simulácie a ich menu sa líši len vizuálne, preto budú jednotlivé polia popísané len pre LTE.

Na vrchu menu v riadku označenom *Eb/N0* sa nachádzajú tri polia, v ktorých je možné zadať hodnoty odstupe signál/šum pre, ktoré bude simulácia prebiehať. Prvé pole určuje počiatočnú hodnotu, v ktorej simulácia začne, naopak druhé pole určuje konečnú hodnotu, kde simulácia skončí. V poslednom treťom poli sa zadáva hodnota kroku medzi minimálnou a maximálnou hodnotou, čím sa určujú ďalšie hodnoty pre simuláciu. S nastavením menšieho kroku dostávame presnejšie grafy, avšak čas simulácie sa výrazne predlži. V druhom riadku s názvom *Data* sa zadáva počet generovaných dát (bitov) pre kodér. S väčším počtom generovaných dát je simulácia presnejšia no opäť sa čas simulácie predlžuje. Tretí riadok s názvom *Blok* obsahuje popupmenu, kde je možné zvoliť povolené veľkosti bloku dát, do ktorého sa rozdelia generované dáta pri kódovaní a následnom dekódovaní. Nasledujúci riadok má názov *Algoritmus*, taktiež ho tvorí popupmenu s možnosťami výberu dekódovacieho algoritmu. Riadok s názvom *Rate* špecifikuje, aký kódový pomer bude použitý v kodéri. To znamená, že koľko výstupných bitov bude pripadať na jeden vstupný bit. Posledný riadok má názov *Iterácie* a určuje počet dekódovacích iterácií dekodéra. V tomto riadku sa tiež nachádza aj checkbox, ktorý v prípade zaškrtnutia zobrazí výsledky všetkých iterácií v opačnom prípade, iba výsledok poslednej. Na spodku menu sa nachádzajú tri tlačítka *Simulovať*, *Uložiť* a *Zahodiť*. Po spustení menu pre vybraný systém je aktívne len tlačítko *Simulovať* a slúži na spustenie simulácie. Akonáhle je simulácia dokončená a zobrazí sa informatívny graf výsledkov simulácie aktivujú sa tlačítka *Uložiť* a *Zahodiť*. Pomocou nich je možné výsledky simulácie buď uložiť do txt súboru alebo vymazať. Užívateľ si musí vybrať jednu z možností, aby mohol pokračovať na ďalšiu simuláciu, pretože v tomto momente je tlačítko *Simulovať* neaktívne, ako aj všetky polia, takže nie je možné meniť parametre simulácie.

V prípade zatvorenia menu pre určitý systém sa automaticky otvorí hlavne menu a opäť je možné zvoliť iný systém, alebo prejsť do menu pre generovanie grafov, ktoré je popísané v nasledujúcej časti kapitoly.



Obr. 4.2: Menu LTE systému.

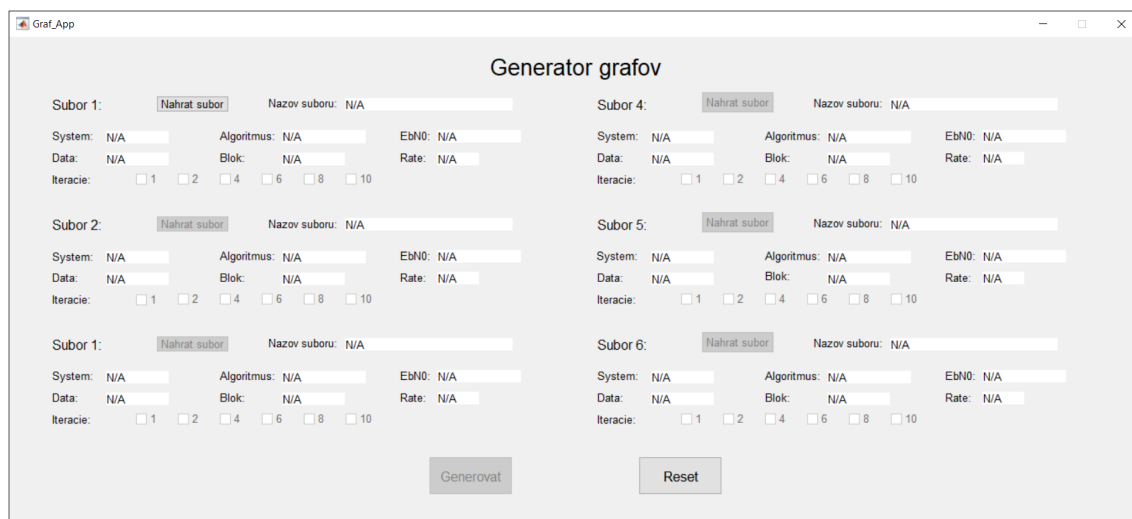
4.3 Menu generovanie grafov

Menu generovanie grafov slúži na vytvorenie grafu z uložených súborov. Do tohoto menu sa môžeme dostať z hlavného menu po kliknutí na tlačítko *Graf* a je zobrazené na obr. 4.3.

V menu je postupne možné vybrať až šesť súborov, z ktorých sa do jednotlivých polí načítajú údaje o simulácii. Nie je potrebné nahráť všetkých šesť súborov, záleží len na užívateľovi, aký graf chce vygenerovať. Generátor povoľuje zvoliť súbory rôznych systémov, dekódovacích algoritmov, alebo blokov dát, avšak vo všetkých simuláciách sa musí zhodovať minimálna hodnota, maximálna hodnota a krok E_b/N_0 . Po načítaní súboru a zobrazení informácií sa aktivujú checkboxy v riadku *Iteracie*, kde je možné zaškrtnutím zvoliť iterácie, ktoré sa zobrazia v grafe. Aktivované sú

len checkboxy tých iterácií, pre ktoré bola simulácia vykonaná. Po zvolení potrebných súborov a vybratí informácií pre graf kliknutím na tlačítko *Generovanie* graf vygenerujeme. Tlačítkom *Reset* je možné resetovať všetky polia a nahraté súbory.

Ako už bolo spomínané pri systémovom menu aj tu sa po zatvorení menu automaticky zobrazí hlavne menu.



Obr. 4.3: Menu generovanie grafov.

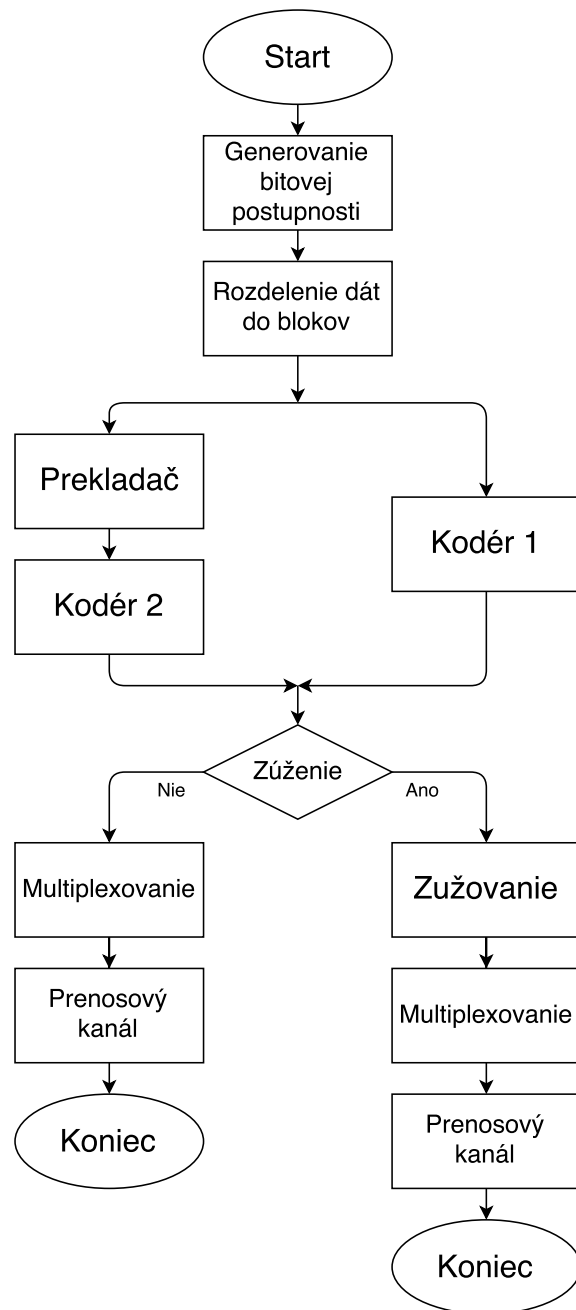
4.4 Blok kódovania

Proces simulácie by sme mohli rozdeliť do dvoch hlavných blokov, bloku kódovania a bloku dekódovania. Obidva tieto bloky využívajú jednu hlavnú funkciu `LTE.m` a navzájom na seba nadväzujú. V tejto časti textu si popíšeme blok kódovania, ktorého vývojový diagram je zobrazený na obr. 4.4 a funkcie, ktoré sú v ňom využívané.

Ako už bolo spomenuté hlavná funkcia, ktorá sa zavolá po stlačení tlačítka *Simulovať* je funkcia `LTE.m`. Tejto funkcii sú predávané všetky parametre simulácie a sú volané ďalšie funkcie, ktoré sa podieľajú na kódovaní. Vstupné premenné funkcii `LTE.m` sú `data`, `blok`, `it`, `it_bool`, `EbN0db`, `rate` a `alg`. Premenná `data` určuje množstvo bitov, ktoré budú generované pre kódér. Druhá premenná `blok` definuje povolené hodnoty blokov dát, čiže veľkosť blokov pre kódér, do ktorých budú rozdelené generované dáta z premennej `data`. Ďalšími premennými sú, premenná `it` určujúca počet dekódovacích iterácií dekódera, premenná `it_bool` určujúca, ktoré iterácie budú zobrazené v grafe a premenná `EbN0db`, čiže hodnoty odstupu signál-šum v ktorých bude simulácia prebiehať. Posledné dve premenné sú `rate`, kódový pomer kódéra a premenná `alg`, ktorá špecifikuje algoritmus použitý pri dekódovaní. Výstupom tejto funkcie je premenná `out1`, chybovosť BER dekódovaných dát a premenná

`out2`, ktorá určuje chybovosť BER nekódovaných dát. Vygenerované dáta sú predané funkcii `LTE_Kod.m`, ktorá zabezpečuje rozdelenie dát do blokov, zúženie a multiplexovanie pre prenosový kanál. Táto funkcia využíva funkciu `LTE_Koder.m`, v ktorej prebieha samotné kódovanie pre jednotlivé kodéry a má vstupné premenné `data` a `c`. V prvej premennej sú funkcii predané generované dáta, rozdelené do požadovaných blokov, ktoré budú zakódované a druhá určuje poradie kodéra. Toto poradie zabraňuje pridávaniu ukončovacích bitov v oboch kodéroch. Funkcia `LTE_Kod.m` má tri vstupné premenné. Vygenerované dáta `data`, veľkosť bloku `b` a kódový pomer kodéra `rate`. Výstupom tejto funkcie sú multiplexované dáta z oboch kodérov, označené ako `out`. Tieto dáta vstupujú do funkcie `PreKanAWGN.m` označené ako `data` spolu s premennou `EbN0db`, určujúcou hodnotu odstup signál/šum a premennou `rate`, ktorá má rovnaký význam ako v predchádzajúcich funkciách. Vo funkcii `PreKanAWGN.m` prebieha simulovanie AWGN kanála a tým pridanie chýb do kódovaných dát. Výstupom sú štyri premenné `L_c` definujúca spoľahlivosť prenosového kanála, `ber` chybovosť prenesených dát, `chyby` počet chýb, ktoré boli pridané do kódovaných dát a premenná `out`, samotné prenesené dáta.

Táto funkcia ukončuje blok kódovania. Proces simulácie pokračuje ďalej vo funkcii `LTE.m`, avšak sa na ňom podieľajú ďalšie funkcie, ktoré sú popísané v nasledujúcej časti kapitoly.



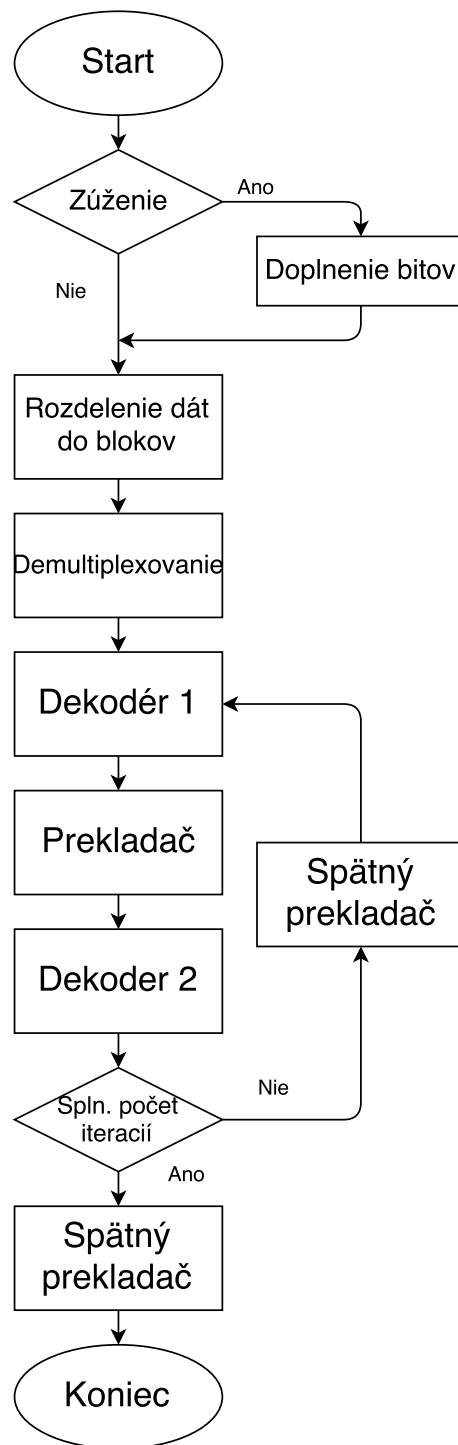
Obr. 4.4: Vývojový diagram kódovania.

4.5 Blok dekódovania

V tejto časti textu je popísaný proces dekódovania a funkcie, ktoré sú pri ňom využívané. Vývojový diagram procesu dekódovania je na obr. 4.5.

Proces simulácie prebieha ďalej vo funkcii `LTE.m` a výstup funkcii `PreKanAWGN.m`, čiže prenesené dáta sú predané funkcii `LTE_Blok.m`, ktorá zabezpečuje rozdelenie dát do požadovaných blokov pre dekódovanie. Vstupmi funkcie sú prenesené dáta `data`, premenná `blok` určujúca veľkosť bloku, do ktorého budú dáta rozdelené a kódový pomer kodéra `rate`. Na výstupe dostaneme dáta rozdelené do blok v premennej `out`. Tieto bloky sú postupne predávané na vstup funkcie `LTE_Dem.m` ako premenná `data`. Funkcia `LTE_Dem.m` slúži na demultiplexovanie blokov do hodnôt reprezentujúcich výstupy jednotlivých kodérov, ktoré sú predávané na výstup `out`. Táto funkcia ma ešte jednu vstupnú premennú a tou je `rate`, čiže kódový pomer. Akonáhle sú dáta rozdelené do blokov a demultiplexované pre dekodéry je na rade proces samotného dekódovania. Tento proces zabezpečuje funkcia `LTE_Dek.m` na vstup, ktorej prichádzajú dáta pre dekodéry `data`, ďalej premenná `L_c` určujúca spoľahlivosť prenosového kanála, premenná `it` čiže počet dekódovacích iterácií a ako posledná premenná `alg` definujúca algoritmus, ktorý bude použitý pri dekódovaní. Táto funkcia využíva funkciu `LTE_Trellis.m`, ktorá slúži na prevod vytváracích polynómov kodéra do trellis diagramu a má jednu vstupnú premennú `g`, čo sú spomínané vytváracie polynómy v binárnom tvare. Na výstupe funkcie dostávame nasledujúce stavy kodéra v trellis diagrame `naStav`, nasledujúce výstupy `naOut`, predchádzajúce stavy `preStav` a poslednú premennú predchádzajúce výstupy `preOut`. V závislosti aký dekódovací algoritmus je zvolený sa použije jedna z nasledujúcich funkcií `LTE_SOVA.m`, `LTE_logMAP.m` alebo `LTE_MAXlogMAP.m`. Jednotlivé funkcie reprezentujú rôzne dekódovacie algoritmy, avšak na vstup každej sú predané rovnaké premenné. Z tohoto dôvodu bude popísané len funkcia `LTE_SOVA.m` na vstup, ktorej prichádzajú výstupné premenné funkcie `LTE_Trellis.m`, čiže `naOut`, `naStav`, `preOut` a `preStav`. Ďalšími vstupnými premennými sú premenná `data`, ktorá určuje dáta pre dekódovanie, premenná `g` definujúca vytváracie polynómy kodéra, premenná `L_uk` určujúca a-priori pravdepodobnosť z predchádzajúceho dekódera a premenná `index` určujúca poradie dekódera. Na výstupe týchto funkcií dostávame a-posteriori pravdepodobnosť `out`, z ktorej následne vo funkcii `LTE_Dek.m` vypočítame a-priori pravdepodobnosť pre ďalší dekoder, alebo pri dosiahnutí poslednej dekódovacej iterácii tvrdý výstup dekódera, z ktorého sa určí dekódovaná výstupná bitová postupnosť a je predaná na výstup funkcie ako premenná `out`.

V ďalšom kroku je dekódovana bitová postupnosť porovnaná s generovanými dátami a tým určená chybovosť BER. Následne je vygenerovaný informatívny graf a podľa voľby užívateľa sú dáta buď uložené, alebo vymazané.



Obr. 4.5: Vývojový diagram dekodovania.

5 SIMULÁCIE

5.1 Úvod

Táto kapitola obsahuje výsledky simulácií vykonaných pre systémy LTE, CCSDS a WiMAX. Vlastnosti týchto systémov sú popísané v kap. 3.1, 3.2, 3.4. Kapitola je rozdelená do piatich častí, vplyv veľkosti bloku na výkonnosť turbo kódov, vplyv zúženia, vplyv dekódovacieho algoritmu, vplyv počtu iterácií a porovnanie výkonnosti simulovaných systémov. Vo všetkých simuláciách bol využívaný AWGN prenosový kanál, BPSK modulácia a objem generovaných dát 1 000 000. Taktiež vo všetkých grafických závislostiach je zobrazená aj chybovosť nekódovaných dát, aby bolo možné porovnať ziskovosť turbo kódov voči nekódovaným dátam. Výkonnosť turbo kódov môže byť ovplyvnená mnohými parametrami. Vo vykonaných simuláciách sme sa zamerali na nasledujúce parametre a ich vplyv:

- veľkosť bloku dát
- zúženie
- počet dekódovacích iterácií
- dekódovací algoritmus
- porovnanie výkonnosti systémov

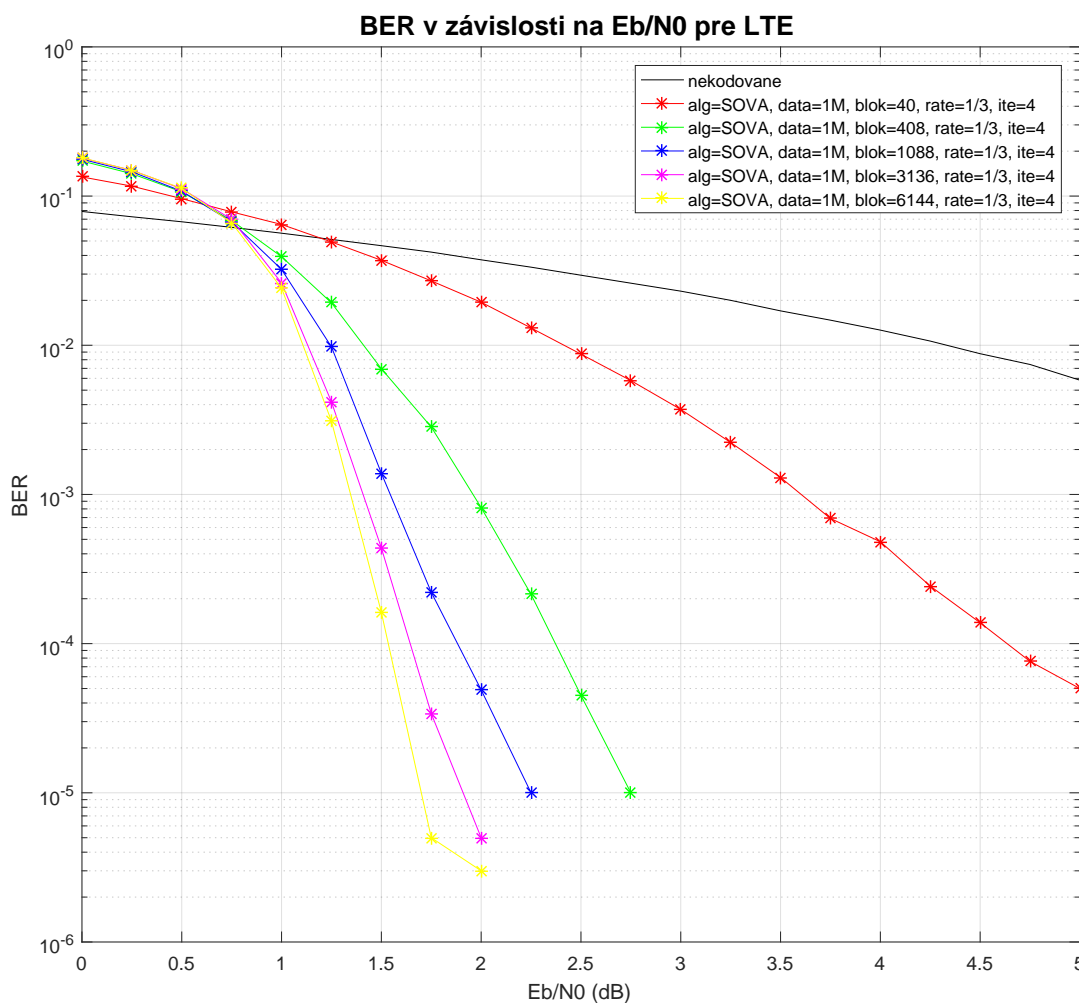
5.2 Vplyv veľkosti bloku

V tab. 5.1 sú uvedené parametre simulácií turbo kódov pre jednotlivé systémy. Veľkosti blokov dát boli volené pre každý systém samostatne a to z hodnôt, ktoré využívajú.

Tab. 5.1: Parametre simulácií - vplyvu veľkosti bloku dát

Kodér	LTE	CCSDS	WiMAX
Vytváracie polynómy	13, 15	23, 33, 25, 37	15, 13, 11
Prekladač	LTE	CCSDS	WiMAX
Gen. dáta	1 000 000	1 000 000	1 000 000
Blok dát	40, 408, 1088, 3136, 6144	1784, 3568, 7136, 6144	12, 72, 192, 576, 1152
Zúženie	1/3	1/6	1/5
Dekódovací algoritmus	SOVA	SOVA	SOVA
Počet iterácií	4	4	4

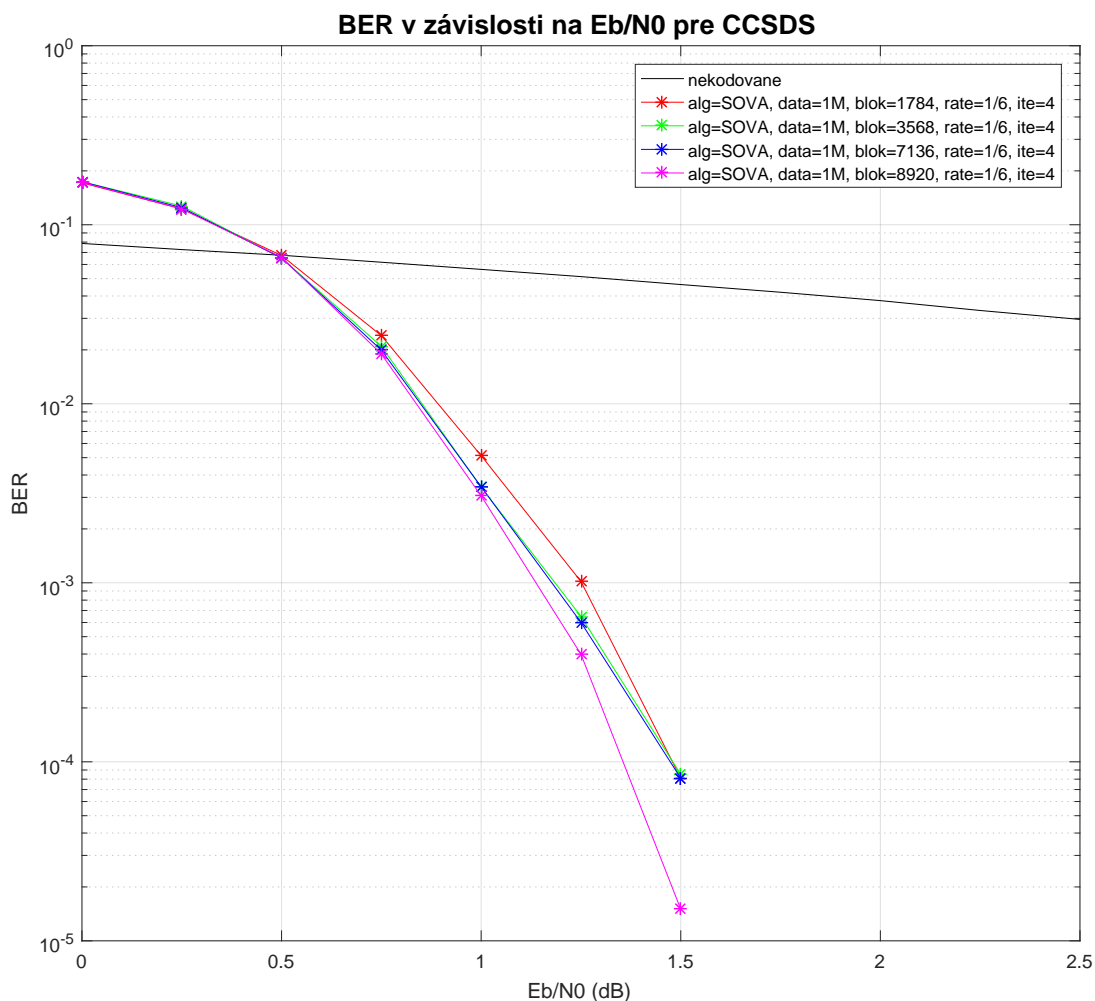
Grafická závislosť na obr. 5.1 zobrazuje výkonnosť turbo kódov v LTE systéme pre rôzne veľkosti blokov dát. Minimálna hodnota bloku dát je 40 bitov, naopak maximálna 6144 bitov. Z grafu je zrejmé, že čím bola veľkosť bloku dát väčšia, tým sa zvyšovala aj výkonnosť turbo kódov. Pri maximálnej a minimálnej hodnote bloku a pri chybovosti BER 10^{-4} , sa dostávame ku kódovému zisku až okolo 3 dB. Ak porovnáme rozdiel v kódovom zisku medzi dvoma minimálnymi hodnotami bloku dát, ktorá bola približne 2,2 dB a dvoma maximálnymi, približne 0,15 dB zisťuje, že pri menších hodnotách bloku dát dostávame oveľa väčší kódový zisk.



Obr. 5.1: Výkonnosť LTE v závislosti na veľkosti bloku.

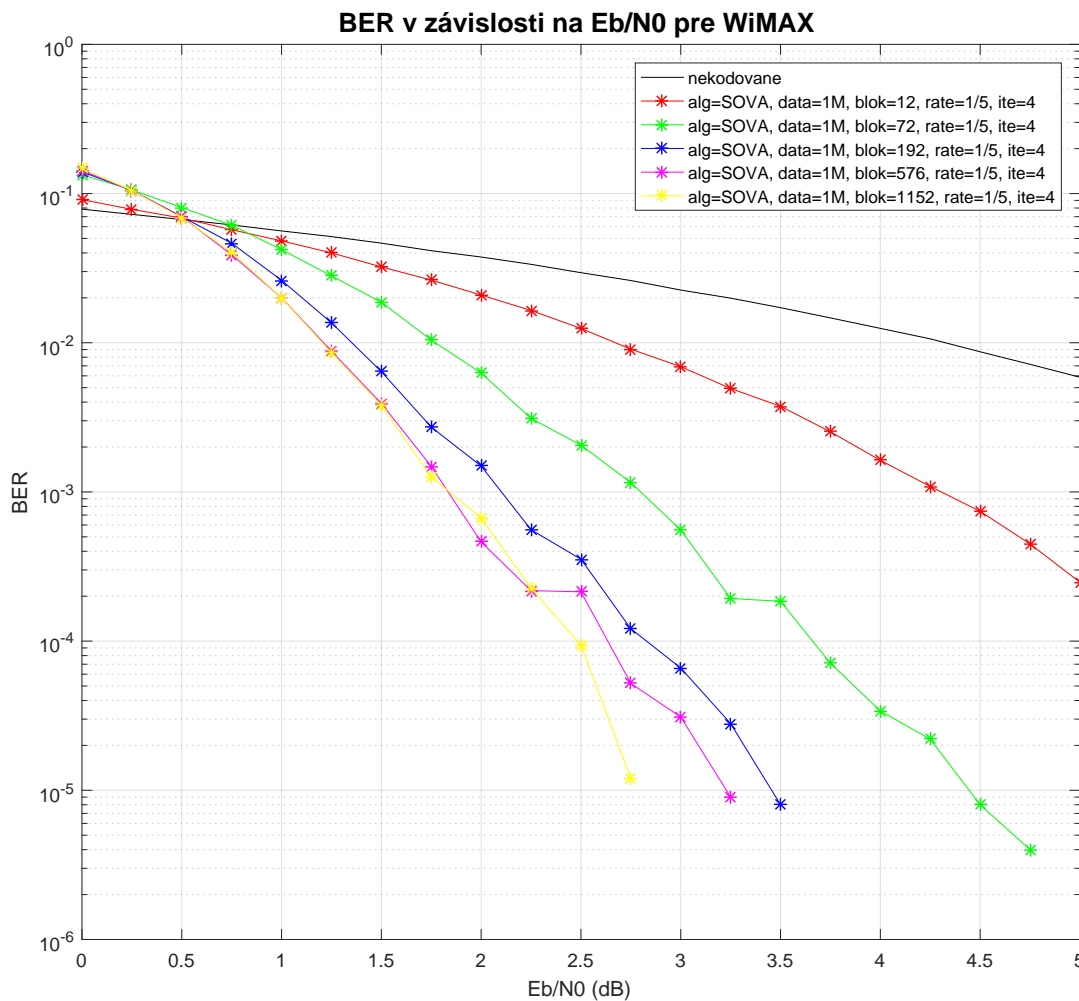
Druhý systém, na ktorý sa pozrieme je systém CCSDS a vplyv veľkosti bloku dát na turbo kód využívaný v tomto systéme. Systém využíva štyri veľkosti bloku dát a grafická závislosť ich vplyvu je zobrazená na obr. 5.2. Z grafu je na prvý pohľad zrejmé, že pri systéme CCSDS tento parameter nehrá rozhodujúcu úlohu. Ziskovosť medzi maximálnou hodnotou veľkosti bloku dát 8920 a minimálnou hodnotou 1784 pri BER 10^{-4} , bola približne 0,1 dB. Dokonca pri dvoch zostávajúcich veľkostiach

blokov dát vidíme prekrývajúce sa krivky, čo značí totožné hodnoty chybovosti BER.



Obr. 5.2: Výkonnosť CCSDS v závislosti na veľkosti bloku.

Posledným testovaným systémom je systém WiMAX. Pre porovnanie výkonnosti turbo kódov v tomto systéme boli vybraté štyri veľkosti blokov dát a ich grafická závislosť je zobrazená na obr. 5.3. Kódový zisk medzi maximálnou hodnotou bloku dát 1152 a minimálnou hodnotou 12, pri chybovosti BER 10^{-3} bola približne 2,65 dB. Opäť ako v prípade systému LTE sme väčšiu ziskovosť turbo kódov dostali pri menších dĺžkach blokov dát. Medzi dvoma najmenšími hodnotami 12 a 72 bol kódový zisk okolo 1,5 dB a naopak pri dvoch najväčších testovaných dĺžkach 1152 a 576, boli hodnoty chybovosti takmer totožné.



Obr. 5.3: Výkonnosť WiMAX v závislosti na veľkosti bloku.

Z výsledkov simulácií môžeme vidieť, že pri všetkých troch systémoch sme dosiahli výrazne vyššiu výkonnosť turbo kódov pri vyšších dĺžkach blokov dát. Ovšem to neznamená, že využívanie menších blokov dát je nevýhodné. Ak zoberieme do úvahy, že s narastajúcou dĺžkou blokov dát rastie aj čas potrebný na dekódovanie týchto blokov, sú pre nás kratšie bloky výhodnejšie vo viacerých prípadoch, ako napr. real-time prenos zvuku, alebo videa a to aj za cenu vyššej chybovosti. Naopak, pri prenose dátových súborov vyžadujeme čo najnižšiu chybovosť a rýchlosť dekódovania nie je rozhodujúca. Z tohoto dôvodu sú pre tento typ prenosu výhodnejšie väčšie dĺžky blokov dát.

5.3 Vplyv zúženia

Tab. 5.2 obsahuje parametre simulácií turbo kódov, ktoré boli použité pri overovaní vplyvu zúženia na jednotlivé systémy. Problematika zúženia je vysvetlená v kap. 2.4.

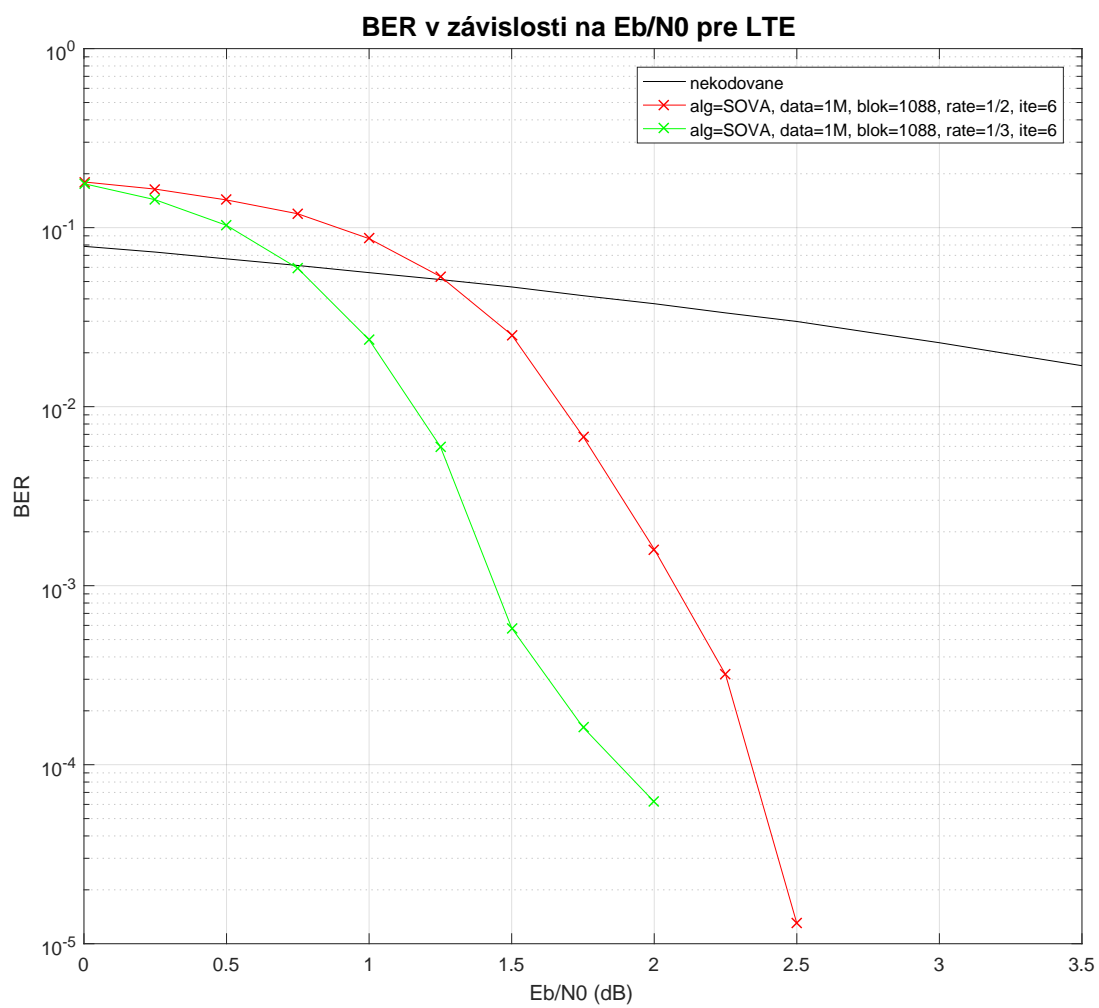
Tab. 5.2: Parametre simulácií - vplyvu zúženia

Kodér	LTE	CCSDS	WiMAX
Vytváracie polynómy	13, 15	23, 33, 25, 37	15, 13, 11
Prekladač	LTE	CCSDS	WiMAX
Gen. dáta	1 000 000	1 000 000	1 000 000
Blok dát	1088	8920	1152
Zúženie	1/3, 1/2	1/6, 1/4, 1/3, 1/2	1/5, 1/3, 1/2
Dekódovací algoritmus	SOVA	SOVA	SOVA
Počet iterácií	6	6	6

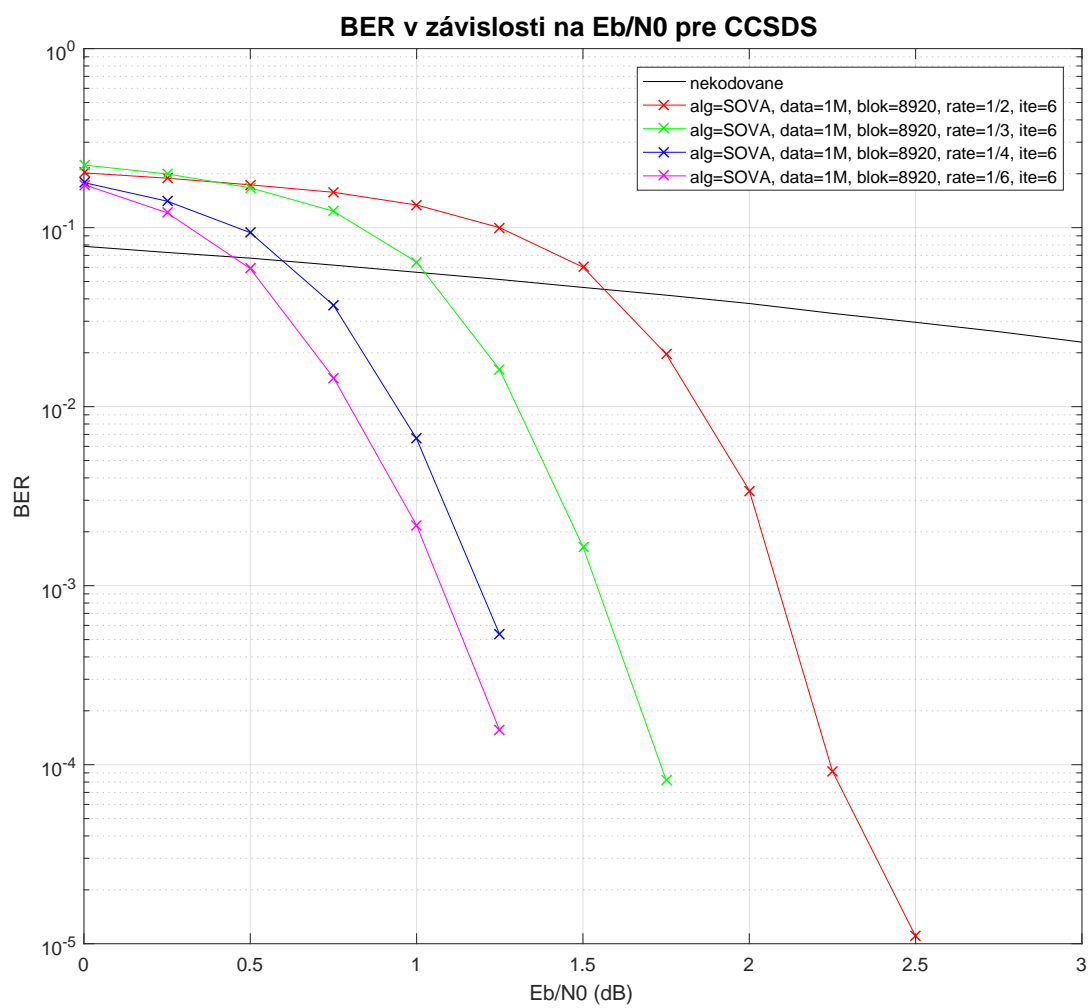
V systéme LTE sa využívajú dva kódové pomery 1/3 a 1/2. Na obr. 5.4 môžeme vidieť vplyv zúženia na výkonnosť turbo kódov v LTE systéme. V prípade využitia zúženia je výkonnosť turbo kódov nižšia v porovnaní s prípadom, kde bol použitý základný kódový pomer. Pri hodnote BER 10^{-3} získame približne 0,7 dB kódový zisk oproti turbo kódu, ktorý použil zúženie na kódový pomer 1/2.

Systém CCSDS obsahuje viacej možností voľby kódového pomeru oproti systému LTE, ich vplyv môžeme vidieť na obr. 5.5. Voľbou kódového pomeru dokážeme ovplyvniť výkonnosť CCSDS systému omnoho viac, ako veľkosťou bloku dát, čo bolo popísané v predchádzajúcej časti tejto kapitoly. Kódový zisk medzi kódovým pomerom 1/6 a 1/2 pri BER 10^{-3} bol približne 1 dB. Ďalej z grafu môžeme usúdiť, že hodnota kódového zisku medzi dvoma nasledujúcimi kódovými pomermi nie je rovnaká. Medzi maximálnym kódovým pomerom 1/6 a nasledujúcou hodnotou 1/4 získavame pri chybovosti BER 10^{-4} okolo 0,15 dB, pri rovnakej chybovosti, avšak medzi hodnotou 1/3 a minimálnou hodnotou 1/2 je kódový zisk približne 0,5 dB.

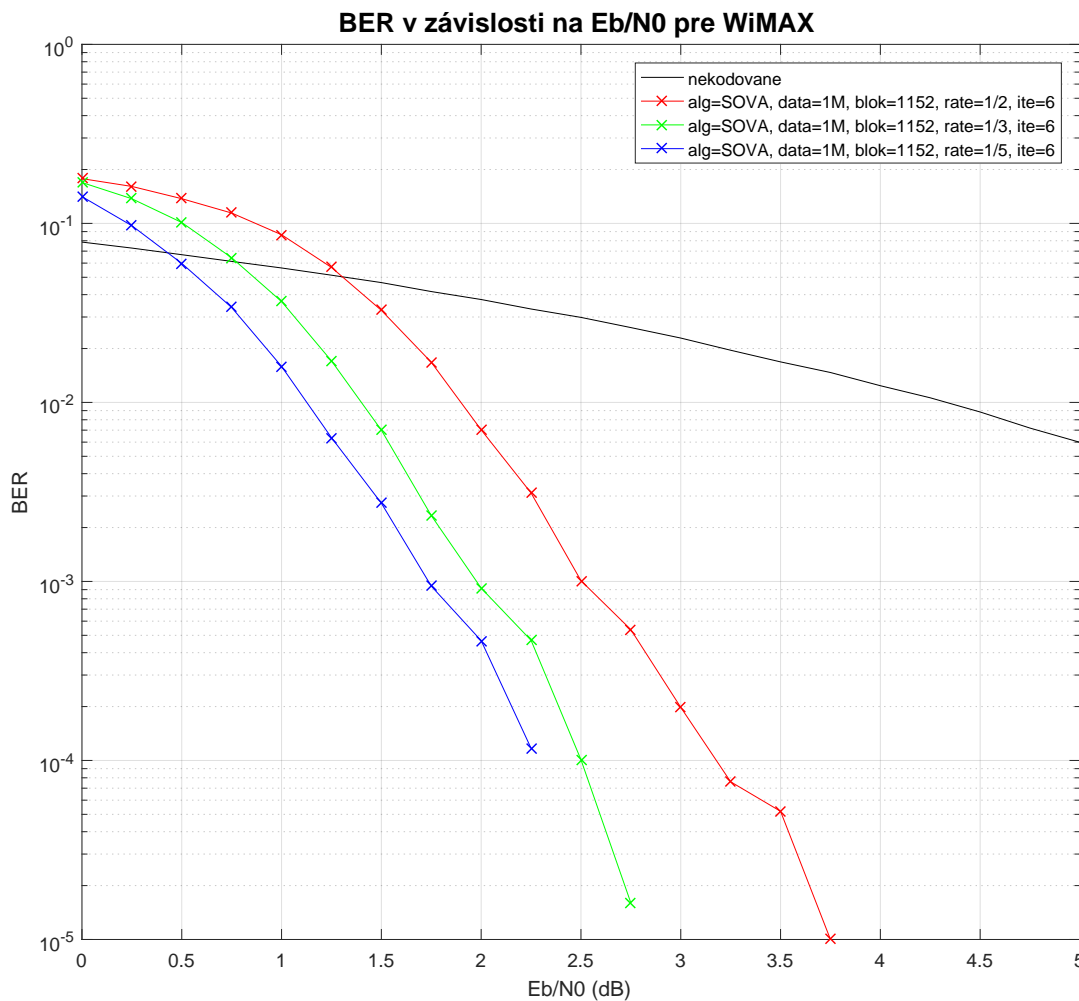
Systém WiMAX využíva tri základné kódové pomery, ktorých hodnoty sú uvedené v tab. 5.2 a ich vplyv na výkonnosť turbo kódov v grafickej závislosti na obr. 5.6. Kódový zisk dosiahnutý medzi nezúženými dátami s kódovým pomerom 1/5 a medzi maximálnym zúžením o hodnote 1/2 sa pri chybovosti BER 10^{-4} pohyboval okolo 1 dB.



Obr. 5.4: Výkonnosť LTE v závislosti na zúžení.



Obr. 5.5: Výkonnosť CCSDS v závislosti na zúžení.



Obr. 5.6: Výkonnosť WiMAX v závislosti na zúžení.

Využívanie zúženia pre nás môže byť výhodné v prípade, kedy chceme znížiť dátový tok a pomocou zúženia môžeme upravovať koľko paritných bitov bude odosielaných cez prenosový kanál. Ďalšou vlastnosťou zúženia je, že nemá vplyv na rýchlosť dekódovania, pretože neprenesené paritné bity sú v dekódery nahradené 0 a dekóduje sa rovnaký počet bitov.

5.4 Vplyv dekódovacieho algoritmu

Parametre simulácií pre jednotlivé systémy môžeme vidieť v tab. 5.3. Popis algoritmov využívaných pri týchto simuláciách nájdeme v kap. 2.5.1 a kap. 2.5.2.

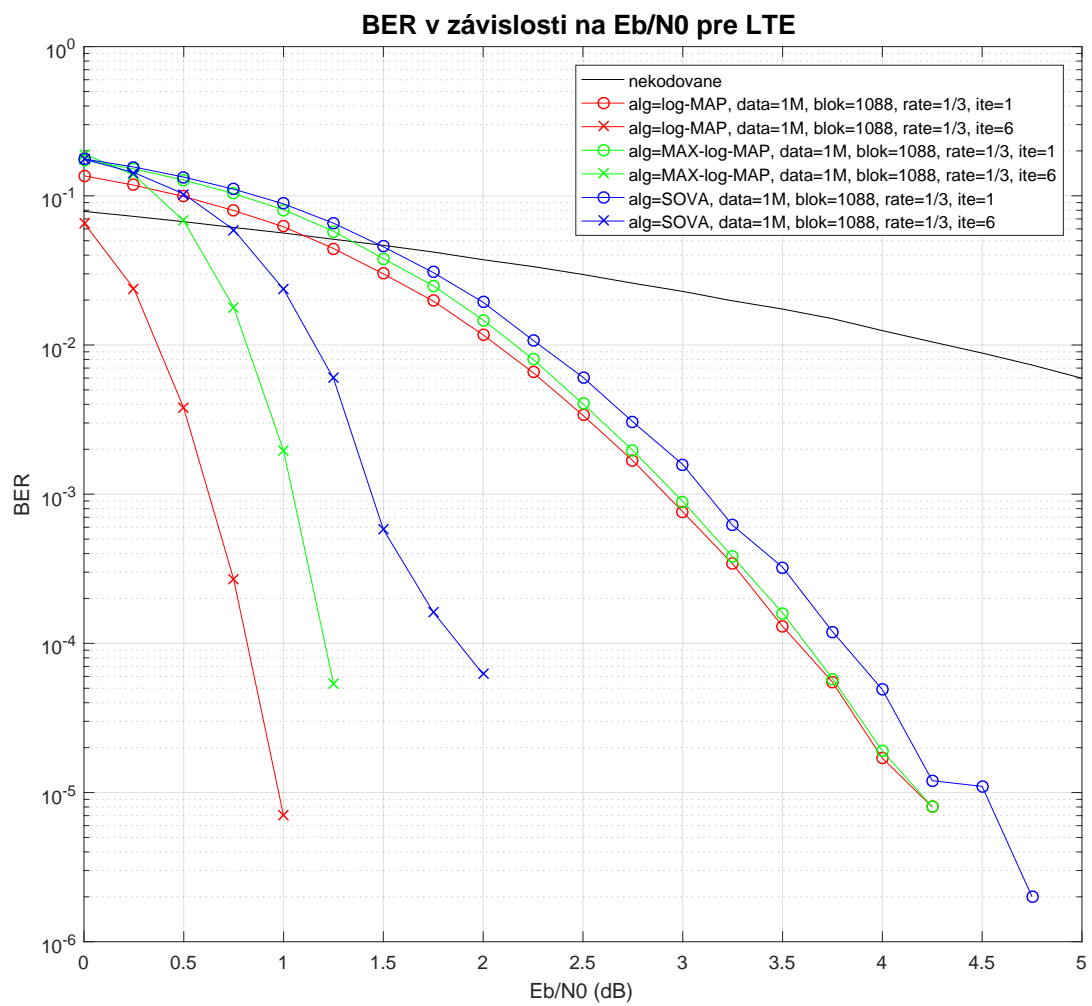
Tab. 5.3: Parametre simulácií - vplyvu dekódovacieho algoritmu

Kodér	LTE	CCSDS	WiMAX
Vytváracie polynómy	13, 15	23, 33, 25, 37	15, 13, 11
Prekladač	LTE	CCSDS	WiMAX
Gen. dáta	1 000 000	1 000 000	1 000 000
Blok dát	1088	8920	1152
Zúženie	1/3	1/6	1/5
Dekódovací algoritmus	SOVA, log-MAP, MAX-log-MAP	SOVA, MAX-log-MAP	SOVA, log-MAP, MAX-log-MAP
Počet iterácií	1, 6	1, 6	1, 6

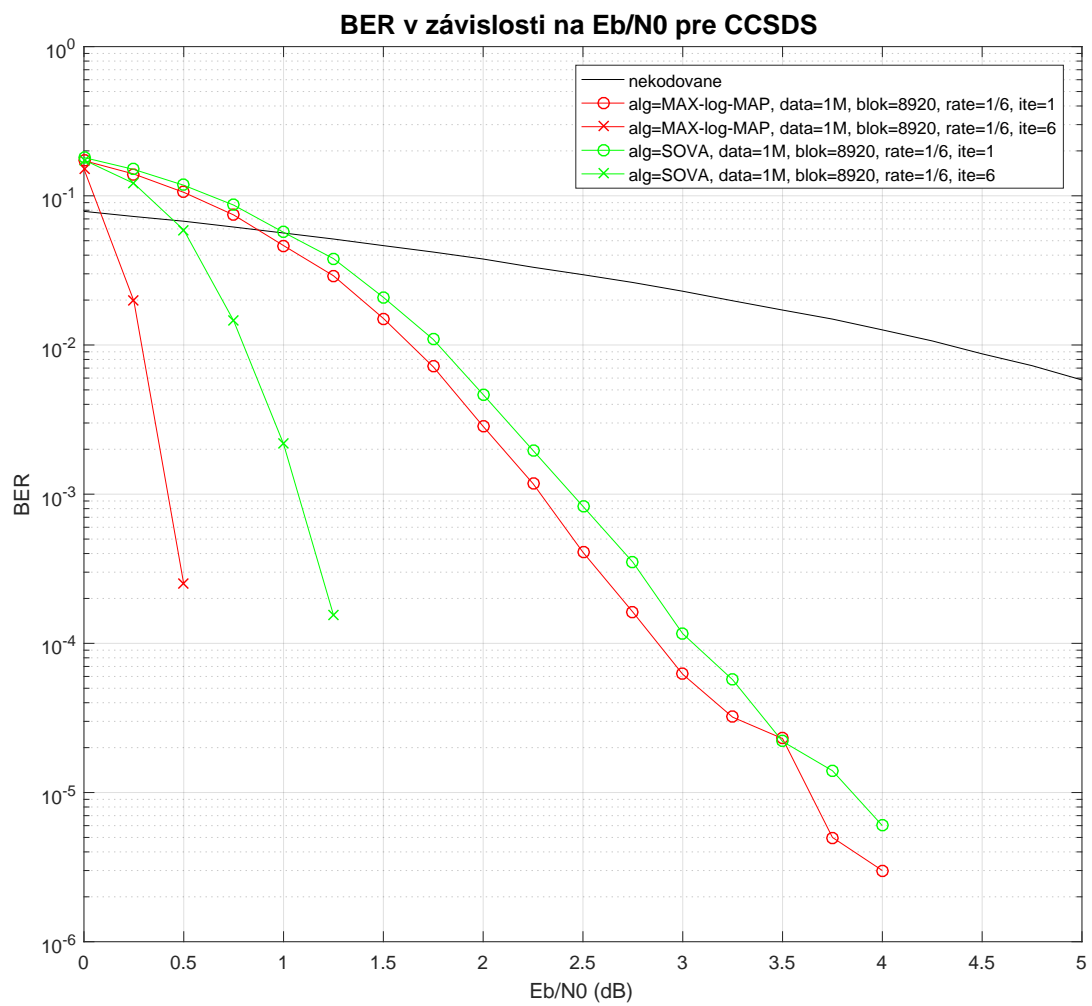
Obr. 5.7 zobrazuje grafickú závislosť výkonnosti turbo kódov v LTE systéme na základe použitého dekódovacieho algoritmu. Z grafickej závislosti vidíme, že najväčšiu výkonnosť sme dosiahli pri algoritme log-MAP a naopak najmenšiu pri algoritme SOVA. Pre porovnanie sú v grafe vynesené výsledky pre rôzne iterácie a z tohoto môžeme usúdiť, že pri väčšom počte iterácií je kódový zisk algoritmu log-MAP ešte väčší. Kým pri porovnaní výkonnosti po prvých iteráciách získavame kódový zisk len okolo 0,25 dB pri chybovosti BER 10^{-4} oproti algoritmu SOVA, tak pri tej istej chybovosti po 6 iterácii sa kódový zisk zvýši až štyri krát na približne 1 dB. Výkonnosť algoritmu MAX-log-MAP sa pohybovala približne v strede medzi dvoma spomínanými algoritmami.

Pre porovnanie vplyvu dekódovacieho algoritmu na výkonnosť turbo kódov využívaných v systéme CCSDS sme použili dva algoritmy a to algoritmus SOVA a algoritmus MAX-log-MAP. Z obr. 5.8 zisťujeme, že pri dekódovaní algoritmom MAX-log-MAP dosiahneme väčšiu výkonnosť. Ako pri predchádzajúcom systéme aj tu sú v grafe uvedené výsledky pre rôzne iterácie a taktiež môžeme zhodnotiť, že s narastajúcim počtom iterácií hrá voľba algoritmu výraznejšiu úlohu. Po prvej dekódovacej iterácii a pri BER 10^{-3} získavame kódový zisk približne 0,2 dB, pri rovnakej chybovosti po 6 iterácii narastá na približne 0,6 dB.

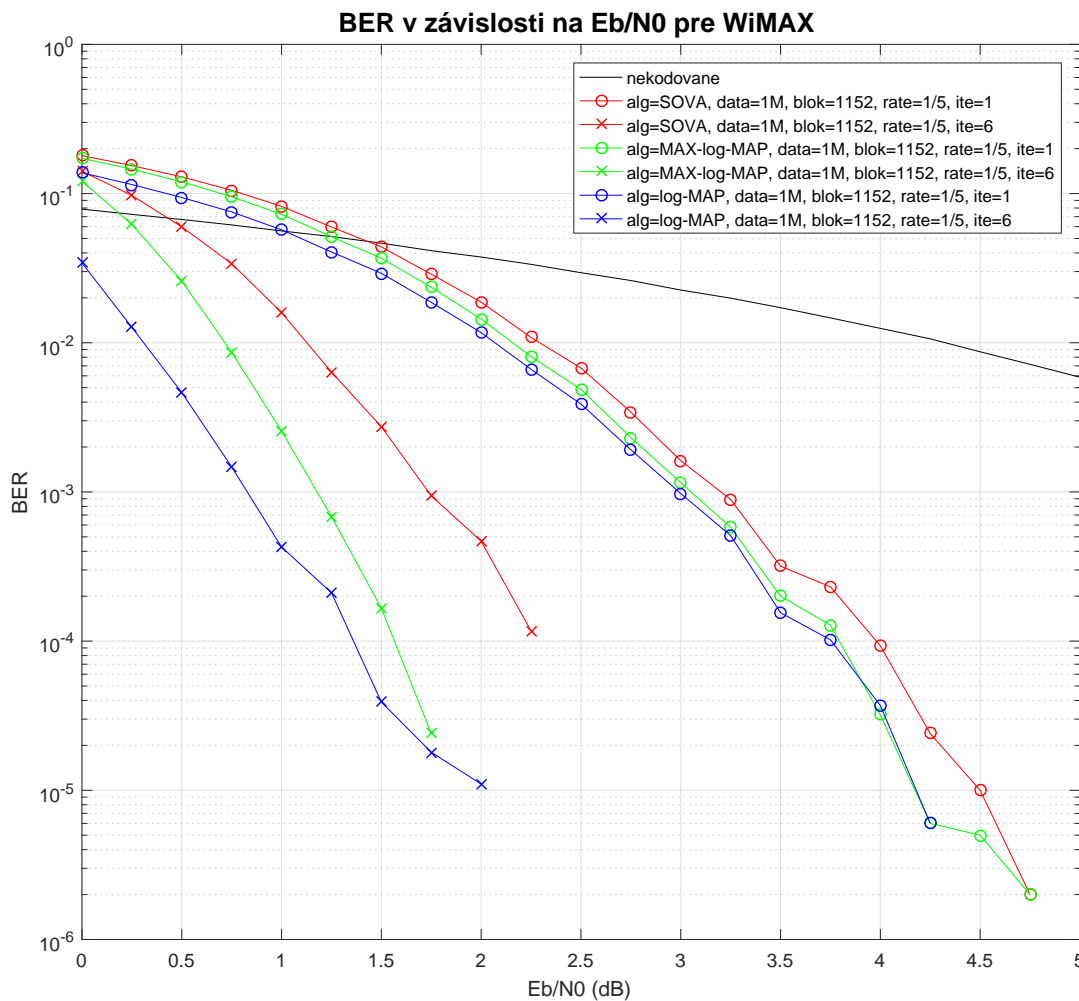
Grafická závislosť na obr. 5.9 zobrazuje vplyv dekódovacieho algoritmu na výkonnosť turbo kódov v systéme WiMAX. Rovnako ako pri systéme LTE boli použité tri dekódovacie algoritmy a aj v tomto prípade dosiahol najväčšiu výkonnosť algoritmus log-MAP. Kódový zisk, ktorý dosiahol algoritmus log-MAP oproti algoritmu SOVA, bol približne 0,25 dB pri prvej iterácii a 1 dB pri iterácii šiestej. Obidve hodnoty boli odmerané pri chybovosti BER 10^{-4} . Porovnaním výsledkov z výsledkami pri systéme LTE, by sme mohli skonštatovať, že vplyv dekódovacieho algoritmu na tieto systémy je takmer identický.



Obr. 5.7: Výkonnosť LTE v závislosti na použitom dekódovacom algoritme.



Obr. 5.8: Výkonnosť CCSDS v závislosti na použitom dekódovacom algoritme.



Obr. 5.9: Výkonnosť WiMAX v závislosti na použitom dekódovacom algoritme.

Ak porovnáme vykonané simulácie zisťujeme, že s algoritmom log-MAP boli dosiahnuté najlepšie výsledky a naopak algoritmus SOVA sa javil ako najhoršia voľba. Z algoritmom MAX-log-MAP sme dosiahli výkonnosť približne v strede medzi spomínanými algoritmami. Za cenu nižšej chybovosti pri algoritme log-MAP dekódovanie turbo kódov z rovnakými parametrami trvá výrazne dlhšie. Keďže je tento algoritmus výpočetne náročný pre potrebu rýchleho dekódovania môže byť algoritmus SOVA vhodnejšia voľba. Algoritmus MAX-log-MAP je podobne ako log-MAP odvodený z algoritmu MAP, no jeho výhoda je menšia výpočetná náročnosť. Z tohto dôvodu ho môžeme považovať za strednú cestu a poskytuje určitý kompromis medzi algoritmom log-MAP a SOVA.

5.5 Vplyv počtu dekódovacích iterácií

Parametre simulácií vykonaných za účelom zistenia vplyvu počtu dekódovacích iterácií na výkonnosť turbo kódov sú zobrazené v tab. 5.4.

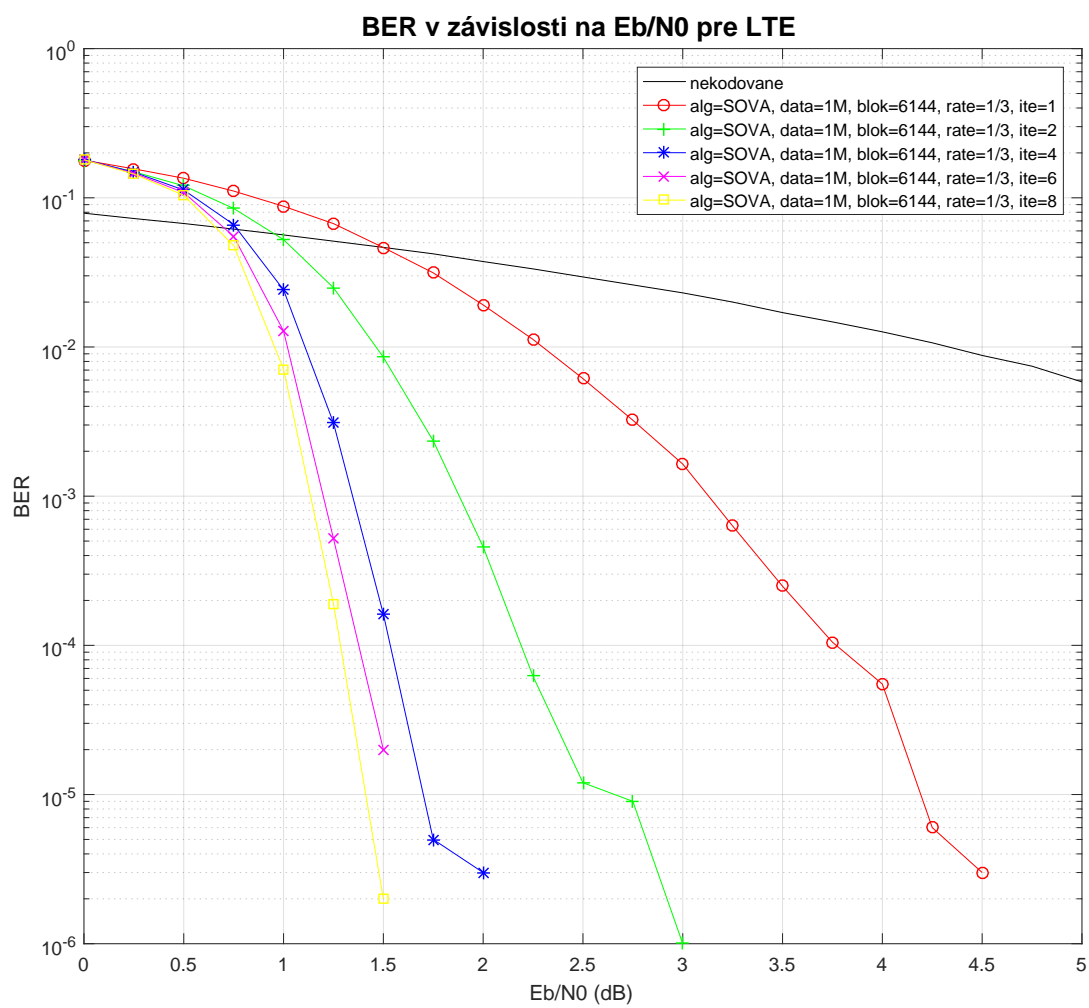
Tab. 5.4: Parametre simulácií - vplyvu počtu dekódovacích iterácií

Kodér	LTE	CCSDS	WiMAX
Vytváracie polynómy	13, 15	23, 33, 25, 37	15, 13, 11
Prekladač	LTE	CCSDS	WiMAX
Gen. dáta	1 000 000	1 000 000	1 000 000
Blok dát	6144	8920	1152
Zúženie	1/3	1/6	1/5
Dekódovací algoritmus	SOVA	SOVA	SOVA
Počet iterácií	1, 2, 4, 6, 8	1, 2, 4, 6, 8	1, 2, 4, 6, 8

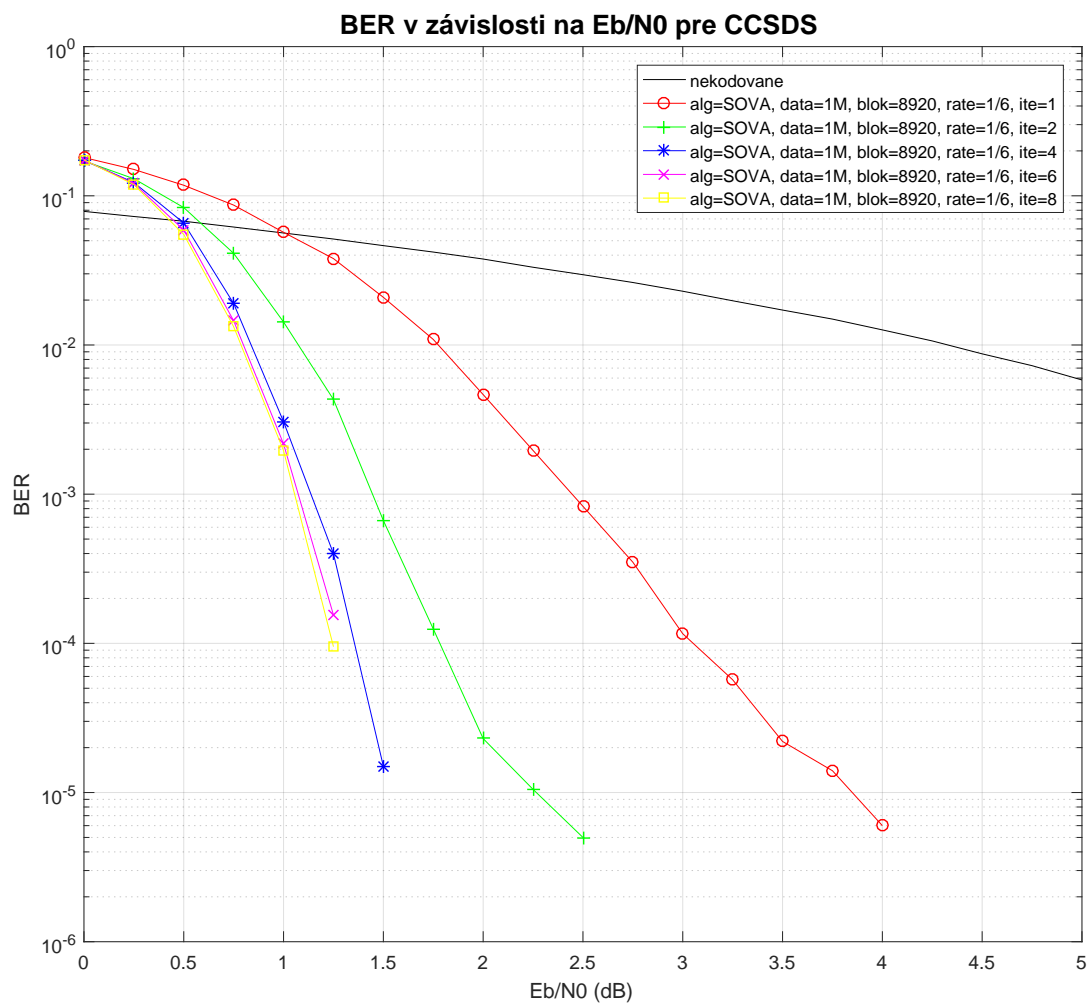
Z grafickej závislosti na obr. 5.10 môžeme vidieť výkonnosť turbo kódov v systéme LTE po jednotlivých dekódovacích iteráciách. S narastajúcim počtom dekódovacích iterácií rastie aj výkonnosť turbo kódov, ovšem medzi nasledujúcimi iteráciami nie je tento nárast rovnaký. Ak porovnáme kódový zisk medzi prvou a druhou iteráciou pri chybovosti BER 10^{-4} , ktorý bol približne 1,55 dB a medzi iteráciami šesť a osem pri rovnakej chybovosti, zistíme, že pri posledných iteráciách bol zisk len okolo 0,1 dB, teda niekoľkonásobne menší.

Na obr. 5.11 vidíme vplyv počtu dekódovacích iterácií na výkonnosť turbo kódov v systéme CCSDS. Ako pri predchádzajúcom systéme LTE, tak aj tu môžeme konštatovať, že s narastajúcim počtom iterácií stúpa výkonnosť turbo kódu a kódový zisk medzi dvoma nasledujúcimi iteráciami nie je rovnaký. Medzi prvou iteráciou a konečnou osmou iteráciou pri chybovosti BER 10^{-5} získavame kódový zisk až 2,8 dB. Pri vyšších iteráciách dostávame opäť niekoľko násobne menší kódový zisk. Medzi prvými dvoma iteráciami pri chybovosti BER 10^{-5} je ziskovosť okolo 1,6 dB a medzi poslednými len 0,1 dB pri rovnakej chybovosti.

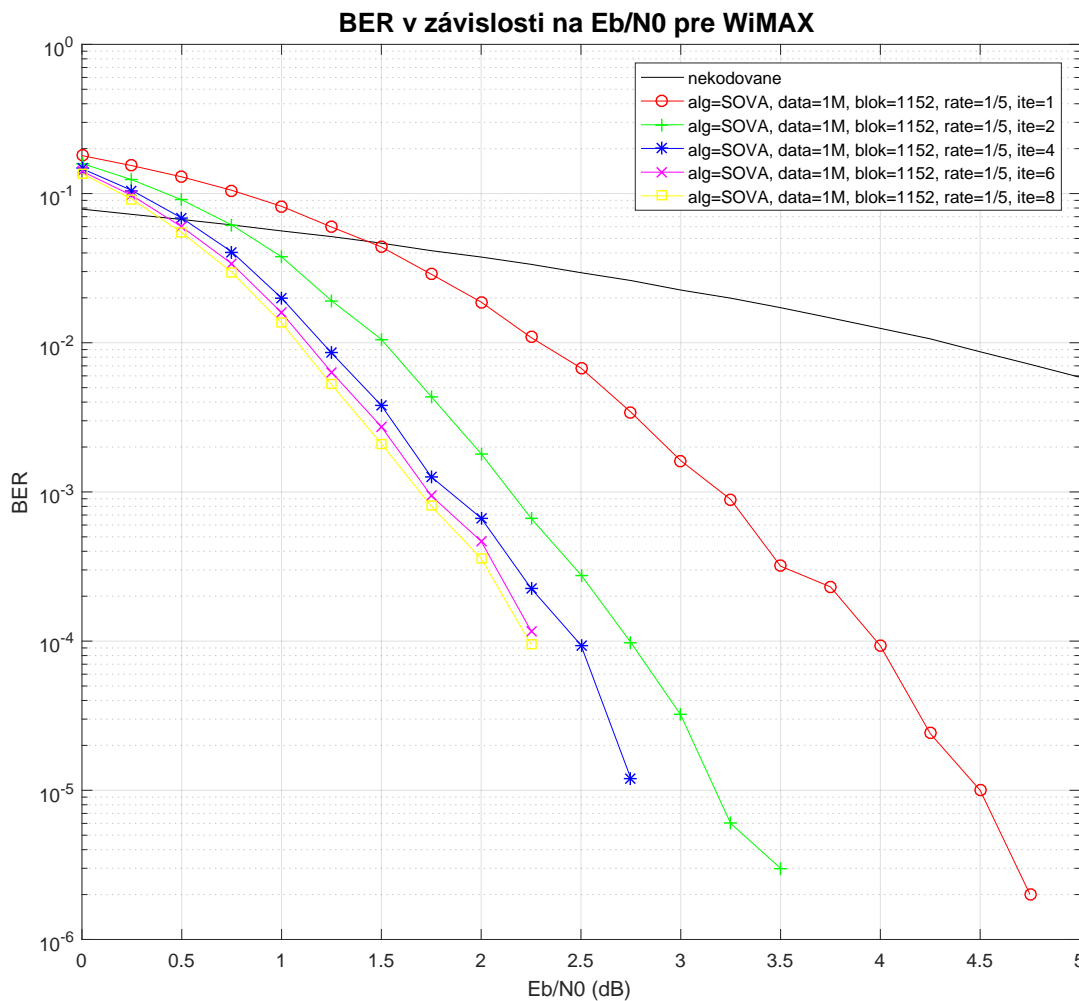
Posledný systém WiMAX a vplyv počtu dekódovacích iterácií na jeho výkonnosť je zobrazený na obr. 5.12. Ako sme očakávali výsledky sú podobné ako v predchádzajúcich systémoch a počet iterácií má rovnaký vplyv. Pri chybovosti BER 10^{-4} , bol kódový zisk medzi prvou a poslednou iteráciou približne 1,75 dB.



Obr. 5.10: Výkonnosť LTE v závislosti na počte dekódovacích iterácií.



Obr. 5.11: Výkonnosť CCSDS v závislosti na počte dekódovacích iterácií.



Obr. 5.12: Výkonnosť WiMAX v závislosti na počte dekódovacích iterácií.

Ako už bolo spomenuté v predchádzajúcom texte, v prípade vyššieho počtu dekódovacích iterácií nedostávame rovnaký kódový zisk medzi dvoma nasledujúcimi iteráciami a čas dekódovania sa výrazne predlžuje. Z tohoto dôvodu je potrebné urobiť kompromis medzi počtom dekódovacích iterácií, chybovosťou, ktorú potrebujeme dosiahnuť a časom, ktorý má dekódovanie trvať, pretože pre rôzny prenos sa môžu tieto požiadavky líšiť.

5.6 Porovnanie výkonnosti systémov

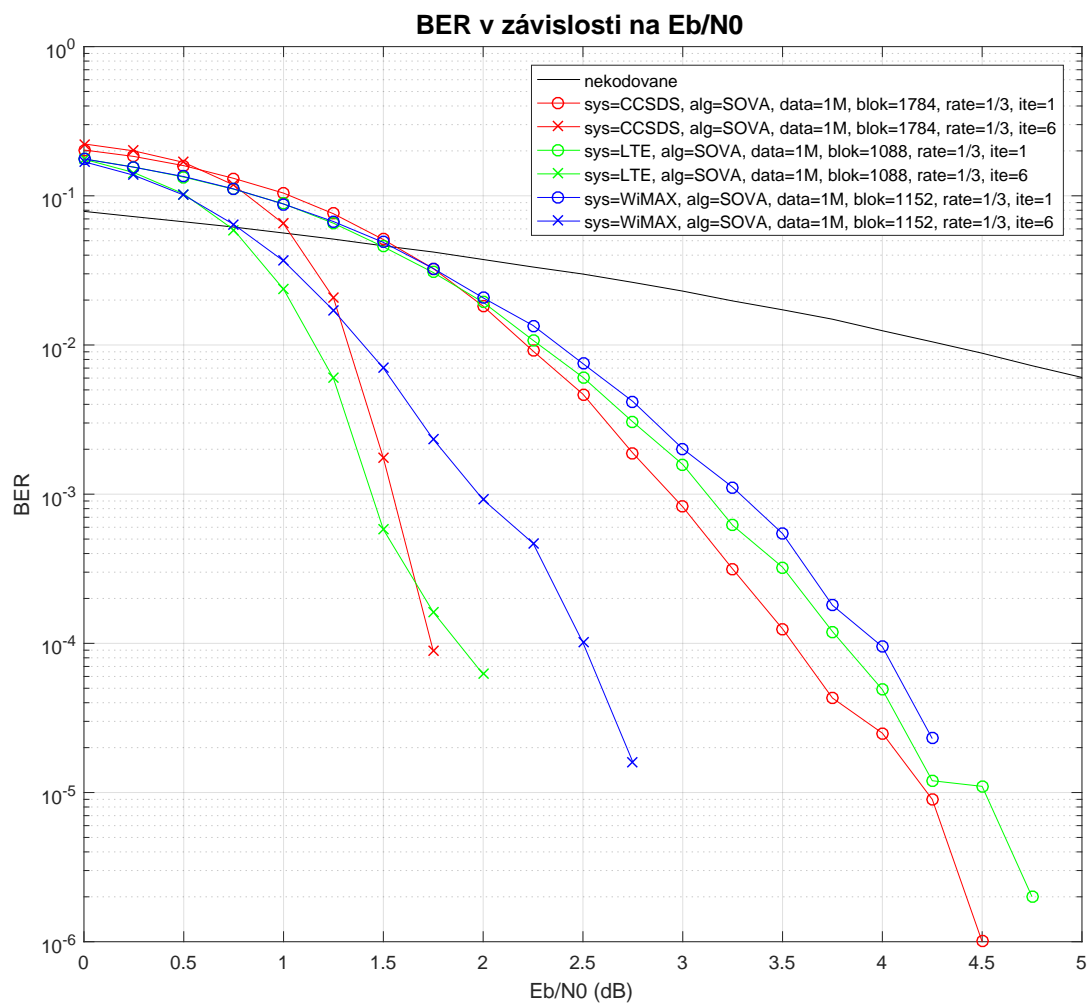
V tejto časti textu si porovnáme výkonnosť turbo kódov v jednotlivých systémoch na základe použitých vytváracích polynómov kodéra. Parametre simulácií sú zobrazené v tab. 5.5. Keďže každý zo systémov používa určité povolené dĺžky blokov dát nebolo možné tento parameter nastaviť na rovnakú hodnotu.

Tab. 5.5: Parametre simulácií - vplyv voľby systému

Kodér	LTE	CCSDS	WiMAX
Vytváracie polynómy	13, 15	23, 33, 25, 37	15, 13, 11
Prekladač	LTE	CCSDS	WiMAX
Gen. dáta	1 000 000	1 000 000	1 000 000
Blok dát	1088	1784	1152
Zúženie	1/3	1/3	1/3
Dekódovací algoritmus	SOVA	SOVA	SOVA
Počet iterácií	1, 6	1, 6	1, 6

Na obr. 5.13 vidíme zobrazené simulácie pre všetky systémy po prvej a šiestej dekódovacej iterácii. Z grafickej závislosti môžeme usúdiť, že najväčšiu výkonnosť sme dostali pri systéme CCSDS. Výsledky môžu byť mierne skreslené, keďže tento systém má zvolený najväčší blok dát, ktorý je ovšem minimálnou hodnotou v tomto systéme. Po prvej dekódovacej iterácii a pri chybovosti BER 10^{-4} sme dostali kódový zisk 0,5 dB oproti systému WiMAX, ktorý sa javil ako najmenej výkonný a približne 0,25 dB oproti systému LTE. Po šiestej dekódovacej iterácii bola situácia podobná a systém CCSDS dosiahol 0,75 dB kódový zisk oproti systému WiMAX a okolo 0,1 dB oproti systému LTE. Ovšem môžeme si všimnúť, že systém CCSDS sa správal rozdielne oproti systémom LTE a WiMAX. Pri vyšších hodnotách chybovosti neplatilo tvrdenie, že systém CCSDS je najvýkonnejší. Ak sa pozrieme na hodnoty pri chybovosti BER 10^{-3} mal systém CCSDS približne o 0,1 dB menší kódový zisk ako systém LTE a dokonca pri chybovosti BER 10^{-2} sa blížil k výkonnosti systému WiMAX.

Z týchto výsledkov môžeme usúdiť, že aj vytváracím polynómom výrazne ovplyvňuje výkonnosť turbo kódov. Keďže systém CCSDS používa väčšie hodnoty polynómov je to pravdepodobne hlavný dôvod získania väčšej výkonnosti turbo kódov v tomto systéme. Ovšem nevýhodou môže byť veľkosť tabuliek pri výpočte metrík v dekóderi, ktoré narastajú s veľkosťou polynómov a môžu výrazne predĺžiť čas dekódovania.



Obr. 5.13: Porovnanie výkonností testovaných systémov.

6 ZÁVER

Bakalársku prácu by sme mohli rozdeliť na dve časti, teoretickú časť a časť praktickú. Cieľom teoretickej časti bakalárskej práce bolo naštudovať problematiku zabezpečovania správ pomocou turbo kódov a venovať sa rôznym prístupom dekódovania týchto správ. Ďalej analyzovať uplatnenie turbo kódov v perspektívnych prenosových technológiách.

V prvej kapitole teoretickej časti tejto práce sú popísané princípy kódovania pomocou konvolučných a blokových kódov, ktorých vlastnosti následne využívajú turbo kódy. Druhá kapitola sa zaoberá samotnými turbo kódmi. Nájdeme v nej vysvetlený postup kódovania, objasnené pojmy ako prekladanie a zúženie a ako posledné popísaný postup dekódovania týchto kódov. V rámci časti turbo dekódovanie sú uvedené a popísané matematické vzťahy a postupy dekódovania pomocou dvoch dekódovacích algoritmov a to algoritmu SOVA a MAP. Tretia kapitola teoretickej časti bakalárskej práce obsahuje uplatnenie turbo kódov v prenosových technológiách. Sú v nej popísané systémy, ktoré tieto kódy využívajú na zabezpečenie správ, ich vlastnosti a parametre, ktoré sú použité pri kódovaní a následnom dekódovaní.

Druhá praktická časť sa zaoberá vytvorením aplikácie v programovom prostredí Matlab slúžiacej na simulovanie a overenie vybraných systémov využívajúcich turbo kódy a ich rôzne prístupy k dekódovaniu zabezpečených správ. Praktická časť je rozdelená do dvoch kapitol. V kapitole s názvom *Implementácia v Matlabe* sú okomentované jednotlivé menu aplikácie a jej funkčnosť. Taktiež táto kapitola obsahuje popísané procesy kódovania a dekódovanie vo vytvorenej aplikácii a úlohu funkcií, ktoré sa podieľajú na týchto procesoch. Posledná kapitola *Simulácie* obsahuje porovnanie a okomentovanie výsledkov simulácií. Simulácie, ktoré boli vytvorené pre potreby tejto bakalárskej práce sú uložené na priloženom CD a je ich viac ako 50. Po porovnaní výsledkov môžeme konštatovať, že výkonnosť turbo kódov je možné ovplyvniť mnohými parametrami, ako napríklad zvýšiť jeho výkonnosť pomocou dĺžky blokov dát, počtom iterácií, alebo použitým dekódovacím algoritmom. Ovšem všetky tieto parametre majú aj negatívny vplyv, ktorým je najmä čas potrebný pre dekódovanie. Preto je potrebné v závislosti na type dát, ktoré sú prenášané, alebo šírke prenosového kanála voliť parametre vhodne.

Hlavným cieľom bakalárskej práce bolo naštudovať si problematiku protichybového zabezpečenia pomocou turbo kódov, oboznámiť sa z ich vlastnosťami a princípmi, avšak treba poznamenať, že táto problematika obsahuje omnoho viac odvodených typov kódov, ktoré však presahujú rozsah tejto bakalárskej práce. Preto verím, že týmto boli splnené všetky body zadania v dostatočnom rozsahu.

LITERATÚRA

- [1] BERROU, Claude, Alain GLAVIEUX a Punya THITIMAJSHIMA. Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1. In: *Proceedings of ICC '93 - IEEE International Conference on Communications* [online]. Geneva: IEEE, 1993, s. 1064-1070 [cit. 2016-11-07]. DOI: 10.1109/ICC.1993.397441. ISBN 0780309502. Dostupné z URL: <<http://ieeexplore.ieee.org/document/397441/>>.
- [2] LIN, Shu a Daniel J. COSTELLO. *Error control coding: fundamentals and applications*. 2nd ed. Upper Saddle River, N.J.: Pearson-Prentice Hall, c2004. ISBN 9780130179739.
- [3] ŠEDÝ, Jakub. Turbo konvoluční kódy. *Electrorevue* [online]. 2011, 13(3), 1-6 [cit. 2016-11-14]. Dostupné z: <<http://www.elektrorevue.cz/cz/download/turbo-konvolucni-kody/>>
- [4] KRAVECOVÁ, Daniela. *Základy kódovania*. Košice: Katedra matematiky a teoretickej informatiky, Fakulta elektrotechniky a informatiky, Technická univerzita v Košiciach, 2012. ISBN 978-80-553-1178-4.
- [5] HANZO, Lajos, T. H. LIEW a B. L. YEAP. *Turbo coding, turbo equalisation, and space-time coding*. Hoboken, N.J.: J. Wiley, c2002. ISBN 0470847263.
- [6] ŠEDÝ, Jakub. *Turbo konvoluční a turbo blokové kódy*. Brno, 2011. Diplomová práce. Vysoké učení technické v Brne. Vedoucí práce Ing. Pavel Šilhavý, Ph.D.
- [7] Turbo code. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2016-12-09]. Dostupné z: <https://en.wikipedia.org/wiki/Turbo_code>
- [8] VAFI, Sina, Tadeusz A. WYSOCKI a Ian BURNET. *Convolutional Interleaver for Unequal Error Protection of Turbo Codes* [online]. USA, Nebraska, 2003 [cit. 2016-12-09]. Dostupné z: <<http://digitalcommons.unl.edu/cgi/viewcontent.cgi?article=1067&context=computerelectronicfacpub>> University of Nebraska - Lincoln.
- [9] DENNETT, Christopher Paul. *An investigation of Turbo Codes over Mobile Wireless Channels*. Wolverhampton, 2006. Disertačná práca. University of Wolverhampton.
- [10] SKLAR, Bernard. *Digital communications: fundamentals and applications*. 2nd ed. Upper Saddle River, N.J.: Prentice-Hall PTR, c2001. ISBN 0130847887.

- [11] KEROUÉDAN, Sylvie a Claude BERROU. Turbo code. In: *Scholarpedia the peer-reviewed open-access encyclopedia* [online]. 2010 [cit. 2016-11-07]. Dostupné z: <http://www.scholarpedia.org/article/Turbo_code>
- [12] *ETSI TS 125 212 V5.1.0 (2002-06): Universal Mobile Telecommunications System (UMTS); Multiplexing and channel coding (FDD)*. Release 5. France: European Telecommunications Standards Institute, 2002.
- [13] *Physical Layer Standard for cdma2000 Spread Spectrum Systems*. Version 1.0. USA: 3rd Generation Partnership Project 2, 2004.
- [14] *ETSI TS 136 212 V10.0.0 (2011-01): LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and channel coding*. Release 10. France: European Telecommunications Standards Institute, 2011.
- [15] *Consultative Committee for Space Data Systems: TM synchronization and channel coding: Blue book*. 1. Washington, DC 20546, USA: National Aeronautics and Space Administration, 2003.
- [16] *ETSI EN 301 790 V1.2.2 (2000-12): Digital Video Broadcasting (DVB), Interaction channel for satellite distribution systems*. V1.2.2. France: European Telecommunications Standards Institute, 2000.
- [17] *ETSI EN 301 958 V1.1.1 (2002-03): Digital Video Broadcasting (DVB), Interaction channel for Digital Terrestrial Television (RCT) incorporating Multiple Access OFDM*. V1.1.1. France: European Telecommunications Standards Institute, 2002.
- [18] SPONSOR, LAN/MAN Standards Committee of the IEEE Computer Society and the IEEE Microwave Theory and Technique. *IEEE standard for local and metropolitan area networks. amendment 2: physical and medium access control layers for combined fixed and mobile operation in licensed bands and corrigendum 1*. New York: Institute of Electrical and Electronics Engineers, 2006. ISBN 0738148571.

ZOZNAM SYMBOLOV, VELIČÍN A SKRATIEK

3G	Third Generation Mobile System
3GPP	Third Generation Partnership Project
3GPP2	Third Generation Partnership Project 2
4G	Fourth Generation Mobile System
alg	dekódovací algoritmus
ARQ	Automatic Repeat-reQuest
AWGN	Additive White Gaussian Noise
BCH	Bose-Chaudhuri-Hocquenghem (code)
BER	Bit Error Rate
blok	veľkosť bloku
BPSK	Binary Phase-Shift Keying
CCSDS	Consultative Committee for Space Data Systems
CDMA	Code Division Multiple Access
data	množstvo generovaných dát
DVB-RCT	Digital Video Broadcasting – Return Channel Terrestrial
DVB-RCS	Digital Video Broadcasting – Return Channel via Satellite
FEC	Forward Error Correstion
GSM	Global System for Mobile communications
GUI	grafické používateľské rozhranie – Graphical User Interface
IEEE	Institute of Electrical and Electronics Engineers
ite	počet dekódovacích iterácií
LLR	Log Likelihood Ratio
LSB	najmenej významný bit – Least Significant Bit
LTE	Long-Term Evolution

MAP	Maximum A-Posteriori (Algorithm)
MSB	najviac významný bit – Most Significant Bit
NRC	Non-Recursive Convolutional (code)
rate	kódový pomer
RSC	Recursive Systematic Convolutional (code)
SISO	Soft-Input Soft-Output (decoder)
SNR	odstup signál/šum – Signal to Noise Ratio
SOVA	Soft-Output Viterbi Algorithm
UMTS	Universal Mobile Telecommunications System
WiMAX	Worldwide Interoperability for Microwave Access
XOR	Exclusive OR
$\alpha_t^{(s)}$	stavová pravdepodobnosť
$\beta_t^{(s)}$	stavová pravdepodobnosť v spätnom smere
D	pamäťové bunky kodéra
dB	decibel
$\gamma_{t-1}^{(s)}$	pravdepodobnosť prechodu
k	počet vstupov kodéra
K	Kódové ohraničenie
$L(u_k)$	a-priori pravdepodobnosť
$L(u_k y)$	a-posteriori pravdepodobnosť
L_c	spoľahlivosť prenosového kanála
$L_c y_{t,1}$	systematická hodnota z prenosového kanála
$L_e(u_k)$	hodnota produkovaná dekóderom
m	počet pamäťových buniek kodéra
$M_t^{(s)}$	hodnota akumulovanej metriky

n	počet výstupných bitov
R	Kódový pomer
$u_t^{(s)}$	vysielaný systematický bit
$x_{t,j}^{(s)}$	vysielaný paritný bit
$y_{t,j}^{(s)}$	prijaté bity

ZOZNAM PRÍLOH

A Obsah priloženého CD

63

A OBSAH PRILOŽENÉHO CD

Priložené CD obsahuje všetky súbory, ktoré boli použité pri vytváraní tejto bakalárskej práce. Zložka `\Application_exe_file` obsahuje skompilovanú verziu aplikácie do *.exe súboru. Pre spustenie tohoto súboru je potrebné mať nainštalovanú určitú verziu Matlabu, alebo balíček MCRInstaller. V zložke `\Application_source_code` sú uložené zdrojové kódy aplikácie. Ďalšia zložka `\Latex` obsahuje zdrojové súbory programu L^AT_EX, v ktorom bola bakalárska práca písaná a taktiež sú tu uložené všetky použité obrázky. Z týchto súborov je vygenerované pdf `\Hlavný_dokument`, čo je samotná bakalárska práca. Posledná zložka `\Simulácie` obsahuje uložené súbory všetkých simulácií, ktoré boli vytvorené pre potreby bakalárskej práce.

